



TAMPEREEN TEKNILLINEN YLIOPISTO  
TAMPERE UNIVERSITY OF TECHNOLOGY

ANTTI RAHIKAINEN

MTV:N AINEISTONSIIRTOYMPÄRISTÖN UUSIMINEN

Diplomityö

Tarkastaja: professori Jarmo Harju  
Tarkastaja ja aihe hyväksytty  
Tieto- ja sähkötekniikan tiedekunta-  
neuvoston kokouksessa 8. huhtikuu-  
ta 2015

## TIIVISTELMÄ

Tampereen teknillinen yliopisto

Signaalinkäsittelyn ja tietoliikennetekniikan koulutusohjelma

**RAHIKAINEN, ANTTI: MTV:N AINEISTONSIIRTOYMPÄRISTÖN UUSIMINEN**

Diplomityö, 46 sivua

Toukokuu 2015

Pääaine: Tietoliikenneverkot ja protokollat

Tarkastaja: professori Jarmo Harju

Avainsanat: aineistonsiirto, videomateriaalinkäsittely, Aspera, orkestrointi

Mediayhtiöt vastaanottavat video- ja kuvamateriaalia useista eri lähteistä, kuten tuotantoyhtiöiltä, mainostoimistoilta sekä muilta sidosryhmiltä. Materiaalin vastaanoton pitää olla käytettävissä 24 tuntia vuorokaudessa, vuoden jokaisena päivänä, johtuen toimijoiden maantieteellisestä sijainnista, jolloin eri aikavyöhykkeet eivät saa vaikuttaa materiaalin vastaanottoon. Samoin aikakriittinen aineisto, kuten uutismateriaali pitää saada viipymättä toimituksen käyttöön.

Saapuvan ja ulospäin lähtevän materiaalin ja aineiston automaattinen käsittely on kriittisessä osassa mediayhtiön toimintaa. Automatisoimalla aineistonsiirtoa voidaan saavuttaa huomattavia etuja, kuten säästää työntekijöiden aikaa esimerkiksi laaduntarkistusten osalta.

Tässä työssä tarkastellaan MTV Oy:n uutta aineistonsiirtoympäristöä, joka on rakennettu täyttämään yllämainitut vaatimukset. Työssä käydään läpi ennalta valitulta ohjelmistotoimittajalta hankittujen sovellusten vaatimukset, käyttöönotto sekä aikaisemmin käytössä olleen ympäristön asteittainen korvaaminen uusilla ratkaisulla.

## ABSTRACT

Tampere University of Technology

Master's Degree Programme in Signal processing and Communications

Engineering

**RAHIKAINEN, ANTTI: RENEWAL OF THE MTV'S MEDIA TRANSFER ENVIRONMENT**

Master of Science Thesis, 46 pages

May 2015

Major: Communication networks and protocols

Examiner: Professor Jarmo Harju

Keywords: media asset processing, media transfer, Aspera, orchestration

Broadcasting companies require stable and always available environment for sending and receiving media material to and from production companies, advertisers and other related groups. Some material is very time critical, like news material which need to be on air within minutes after receiving the media asset.

Incoming and outgoing material processing must be automated in order to achieve fast and stable processing for media assets. Quality control and meta-information checks must be executed for each incoming media file. Doing this by hand takes a lot of time and consumes valuable time of the reporters and media asset managers.

This study examines the renewal of the Finnish broadcasting company MTV's media transfer environment in order to achieve stable and efficient environment, which can handle significant amount of incoming material and execute pre-defined workflows to make the material available for broadcasting or published on the website.

## ALKUSANAT

Tämä diplomityö on tehty osana MTV:n tarvetta uudistaa käytössä ollut aineistonsiirto-ympäristö. Olin mukana suunnittelemassa ja toteuttamassa ympäristöä heti alkumetreistä lähtien ja ympäristön uusimisen aikana mieleeni tuli kirjoittaa aiheesta myös diplomityö.

Kiitän ohjaajiani Markus Linnalaa ja Peter Karjalaista jotka auttoivat suuresti työn teknisissä ja kielellisissä asioissa. Samoin kiitän tarkastajaa professori Jarmo Harjua, jonka hyvistä kommentteista sai lisänäkökulmia työlle.

Erityiskiitos lähtee myös Jonne Köpmanille, joka MTV:n puolesta mahdollisti työn kirjoittamisen ja antoi luvan työlle.

Samoin lämmin kiitos lähtee jokaiselle TeLE:n kiltahuoneelle eksyneelle TeLE:läiselle. Ilman TeLE:ä ei olisi tätäkään diplomityötä!

Helsingissä, 10.5.2015

Antti Rahikainen

# SISÄLLYSLUETTELO

<b>1</b>	<b>Johdanto.....</b>	<b>1</b>
<b>2</b>	<b>Aineistonsiirtoympäristön sovellukset .....</b>	<b>3</b>
2.1	Tiedonsiirtoprotokollat .....	3
2.1.1	TCP-pohjaiset siirtoprotokollat .....	4
2.1.2	FASP .....	5
2.1.3	FASP- ja FTP-siirtojen vertailua.....	5
2.2	Aspera Enterprise ja Connect Server .....	7
2.3	Aspera Orchestrator.....	10
2.4	Aspera Faspex.....	13
2.5	Aspera Console .....	16
<b>3</b>	<b>lähtö- ja tavoitetilanne .....</b>	<b>17</b>
3.1	Lähtötilanteen ongelmat .....	18
3.2	Tavoitetilanne.....	19
<b>4</b>	<b>Sovellusten asennus ja rajapinnat .....</b>	<b>22</b>
4.1	Verkkoympäristö .....	22
4.2	Rajapinnat eri järjestelmiin .....	24
4.3	Aspera Connect Server sovelluksen asennus .....	26
4.4	Aspera Orchestrator.....	27
<b>5</b>	<b>Työnkulkujen rakentaminen .....</b>	<b>28</b>
5.1	Bash-skriptien siirto työkuluiksi.....	28
5.2	Työnkulun rakentaminen Orchestrator-sovelluksessa .....	31
<b>6</b>	<b>Ohjelmamateriaalin käsittelyn uusiminen.....</b>	<b>33</b>
6.1	Alkuperäinen toteutus.....	33
6.2	Uuden toteutuksen ensimmäinen vaihe .....	34
6.3	Lopullinen toteutus.....	36
6.3.1	Tiedoston saapuminen ja laadunvarmistus.....	37
6.3.2	Vantage-transkoodaus ja virheen käsittely .....	40
<b>7</b>	<b>Yhteenveto .....</b>	<b>44</b>
	<b>Lähteet.....</b>	<b>46</b>

## LYHENTEET JA MERKINNÄT

RTT	Round-Trip time, aika joka kuluu yhden paketin kuljetessa kohde-palvelimelle ja takaisin.
FTP	File Transfer Protocol, yleinen tiedostonsiirtoprotokolla.
FASP	Fast and secure protocol. Asperan kehittämä tiedostonsiirtoprotokolla.
WAN	Wide Area Network, verkko jossa kohde ja lähdepää sijaitsevat maantieteellisesti kaukana toisistaan.
LDAP	Lightweight Directory Access Protocol, avoin hakemistoprotokolla, jota käytetään esimerkiksi käyttäjähallintaan.
AMQP	Advanced Message Queuing Protocol, viestinvälitys protokolla, jonka avulla voidaan lähettää ja vastaanottaa viestejä hajautettujen järjestelmien välillä.
XSLT	Extensible Stylesheet Language Transformations, ohjelmointikieli jonka avulla voidaan muuntaa XML-dokumentteja uusiksi XML-dokumenteiksi.
REST	Representational State Transfer. Arkkitehtuurimalli HTTP-pohjaisten rajapintojen toteuttamiseen.
NFS	Network File System, *nix-käyttöjärjestelmille kehitetty protokolla, jonka avulla voidaan jakaa tiedostojärjestelmiä lähiverkossa.
CIFS	Common Internet File System, Microsoftin kehittämä tiedostojen ja hakemistojen jakamiseen tarkoitettu protokolla.
Wireshark	Sovellus, jolla voidaan tarkastella verkkoliikennettä.
Mediainfo	Sovellus, jolla voidaan lukea mediatiedoston metatietoja

# 1 JOHDANTO

MTV on suomalainen mediatyhtiö, joka tuottaa uutisia sekä lähettää ohjelmia televisiossa sekä Internet TV-palveluissa ympäri vuorokauden.

Eri medioissa näytettävä sisältö tuotetaan yleisesti yrityksen ulkopuolella eri tuotantoyhtiöiden toimesta ja toimitetaan eri medioiden kautta MTV:n käyttöön. Eri toimijoita on runsaasti ja jokaisella toimijalla on ollut erilaiset tavat toimittaa sisältöä. Käytettyjä sovelluksia ovat olleet esimerkiksi FTP-palvelimet, Aspera Connect Server, sähköposti, sekä fyysiset toimitukset esimerkiksi USB-kovalevyn avulla. Materiaalin vastaanoton sekä lähetyksen on toimittava vuorokauden ympäri, sillä materiaalin tuottajat sekä toimittajat ovat sijoittuneet ympäri maailmaa. Tämän takia aikavyöhyke-erot eivät saa vaikuttaa materiaalin vastaanottoon. Uutismateriaaliksi tarkoitetut toimitukset ovat lisäksi erittäin aikakriittisiä. Tämä asettaa aineistonsiirtoympäristölle tiukat vaatimukset saatavuuden sekä vikasietoisuuden osalta.

Aikaisempi ympäristö oli toteutettu pääosin Bash-skriptien avulla, jotka suorittivat materiaalin siirtoa ulkoverkosta sisäverkkoon. MTV:n tavoitteena oli yhdenmukaistaa eri toimitusmenetelmät siten, että kaikki aineisto sisään ja ulospäin kulkee saman järjestelmän läpi. Projektin tarkoituksena oli korvata vanhan ympäristön skriptit kaupallisella orkestrointisovelluksella, joka samalla mahdollistaisi sisään- ja ulospäin liikkuvalla materiaalille paremman laaduntarkkailun sekä virheen käsittelyn. MTV:n tavoitteena oli myös ulkoistaa järjestelmän ylläpito sekä valvonta sellaiselle taholle, joka pystyisi valvomaan järjestelmän toimintaa ympäri vuorokauden.

MTV valitsi alustan uudistamisprojektin toimittajaksi Cybercom Finlandin, jonka kanssa MTV on tehnyt valmiiksi paljon yhteistyötä. MTV oli tehnyt ennen projektin aloitusta päätöksen ottaa käyttöön Asperan tarjoamia sovelluksia laajemmin kuin aikaisemmassa ympäristössä. Samoin ympäristön määrittely tuli pääosin MTV:n puolelta. Cybercomin projektitiimin tehtäväksi jäi toteuttaa sovellusten asentaminen uuteen ympäristöön, sekä aineistonsiirrossa käytettyjen skriptien siirto uuteen orkestrointisovellukseen.

Tässä diplomityössä tarkastellaan vanhan ja uuden ympäristön arkkitehtuuria sekä käydään läpi uudessa ympäristössä käyttöönotettuja uusia sovelluksia. Samoin tarkastellaan lähemmin kuinka uuden orkestrointisovelluksen käyttöönotto on mahdollistanut uusien työkulkujen toteuttamisen, paremman valvonnan, virheen käsittelyn sekä toiminnallisuuden yleisen parantamisen.

Työn luvussa kaksi käydään läpi ympäristössä käytettävät sovellukset sekä esitellään niiden toiminnallisuuksia. Luvussa kolme esitellään aineistonsiirtoympäristön lähtö- ja

tavoitetilanne. Luvussa neljä käsitellään ympäristön arkkitehtuuria sekä eri rajapintoja, joihin ympäristö on yhteydessä. Luvussa viisi käydään läpi vanhan aineistonsiirtoympäristön skriptien siirto uuden orkestrointisovelluksen päälle sekä esitellään orkestrointisovelluksen toimintaa tarkemmin. Luvussa kuusi käydään läpi projektin tuotoksia sekä tehdään yhteenvetoa siitä miten uusi ympäristö on vastannut alkuperäistä suunnitelmaa.



## 2 AINEISTONSIIRTOYMPÄRISTÖN SOVELLUKSET

MTV oli projektin alkaessa valinnut ympäristön toteutukseen käytettävät sovellukset. Valinta oli kohdistunut Asperasoft-yrityksen tuotteisiin. Valinta kohdistui Asperaan, sillä Asperan tuotteet ovat käytössä lähes kaikilla suurilla tuotantoyhtiöillä kautta maailman ja tuotantoyhtiöt eivät yleisesti tarjoa muita mahdollisuuksia ladata materiaalia Internetin yli muilla kuin Asperan sovelluksilla.

MTV:llä oli ollut ennen projektin alkamista Asperan sovelluksista käytössä ainoastaan yksinkertaiseen tiedostonsiirtoon käytettävä sovellus, joka mahdollisti tiedostojen lataamisen ja vastaanottamisen tuotantoyhtiöiltä. Uuden ympäristön tarkoituksena oli tuoda MTV:n omaan ympäristöön uusia sovelluksia, jotka mahdollistaisivat tehokkaamman materiaalinkäsittelyn ulos- ja sisäänpäin. MTV:llä oli myös tarpeita kehittää yrityksen sisällä tapahtuvaa aineistonsiirtoa uusien sovelluksien avulla ja korvata jaetut verkkolevyt uusilla toteutuksilla ja tehostaa materiaalinkäsittelyn automaatiota ja läpinäkyvyyttä loppukäyttäjille.

### 2.1 Tiedonsiirtoprotokollat

Perinteisesti tiedostonsiirroissa käytetään FTP- tai SFTP-protokollia. Molemmat perustuvat TCP-protokollaan ja ovat OSI-mallissa sovelluserroksella.

Kerros	Esimerkki sovellutuksesta
Sovellus	FTP
Esitystapa	JPG
Yhteysjakso	NFS, NetBios
Kuljetus	TCP
Verkko	IP
Siirtoyhteys	Ethernet
Fyysinen kerros	Ethernet

*Kuva 1: OSI-Malli*

### 2.1.1 TCP-pohjaiset siirtoprotokollat

TCP-pohjaiset tiedostonsiirtomekanismit kuten FTP, HTTP ja rsync ovat toimivia ja luotettavia silloin, kun yhteys palvelimen ja käyttäjän välillä on hyvä ja häiriötön. Tarkoittaen sitä, että verkossa ei tapahdu pakettien hukkumista ja RTT on pieni. TCP:n perusominaisuus on, että jokainen saapunut paketti kuitataan vastaanottajan toimesta, jotta lähettäjä tietää, että lähetetyt paketit ovat varmasti saapuneet perille.

RTT:n kasvaessa TCP-yhteyden siirtokapasiteetti pienenee, sillä lähettäjä joutuu odotamaan kauemmin jo lähetettyjen pakettien kuittausta, ennen kuin se lähettää uuden paketin. Vastaavasti, jos paketteja hukkuu matkalla, eli lähettäjä ei saa kuittausta vastaanottajalta, lähettäjä pienentää lähetysikkunaa. Lähetysikkunalla tarkoitetaan sitä tavumäärää, mikä on yksittäisen paketin koko lähetävässä päässä. Vastaavasti vastaanottavassa päässä on vastaanottoikkuna, joka kuvaa sitä tavumäärää, minkä vastaanottaja pystyy käsittelemään kerralla.

TCP-yhteyden siirtokapasiteetti on siis riippuvainen RTT:stä, lähetys- ja vastaanottoikkunoiden koosta sekä pakettien hukkumistodennäköisyydestä. TCP-yhteyden keskimääräinen siirtokapasiteetti voidaan laskea seuraavan kaavan avulla [1]

Kaava 1: 
$$\frac{1,22 * MSS}{RTT * \sqrt{L}} = \text{Siirron keskimääräinen nopeus}$$

missä MSS on yksittäisen segmentin maksimikoko (yleisesti 1500 tavua), RTT on kiertoaika ja L yksittäisen paketin hukkumistodennäköisyys.

Kaavan mukaan halutaan saavuttaa tämän päivän yritysliittymissäkin yleinen 1 Gbps tiedonsiirtonopeus, ja oletetaan, että RTT on 300 ms. 300 ms on valittu RTT:n arvoksi sillä perusteella, että se vastaa likimain todellista RTT:tä kun lähdepalvelin sijaitsee Los Angelesissa ja asiakasohjelma Suomessa. Tämä vastaa todellista käyttötilannetta, sillä osa tuotantoyhtiöistä sijaitsee Hollywoodissa. Tästä saadaan hyväksytyksi hukkumistodennäköisyydeksi:

$$L = \left( \frac{1,22 * (1500 * 8)b}{0,3s * (1 * 10^9)bps} \right)^2$$

Tästä saadaan L:n arvoksi 0,00000000238, eli noin 1 paketti jokaista 400 000 000 pakettia kohden saa hukkuu matkalla. Tämä kuvastaa hyvin TCP-pohjaisten siirtoprotokollien, kuten FTP:n ongelmia WAN-verkkojen yli tapahtuvissa siirroissa, varsinkin jos jokin linkki matkalla on huonolaatuinen ja pakettien hukkumista tapahtuu. Teoreettiseen maksimiin on erittäin hankala päästä, sillä WAN-verkkojen luotettavuutta ei voida taata kahden eri toimijan välillä.

### 2.1.2 FASP

FASP (Fast and secure protocol) on Asperan kehittämä tiedonsiirtoprotokolla [2]. FASP-protokolla mahdollistaa tiedostojen siirtämisen Internetin yli tarjoten samalla siirrossa tapahtuvien virheiden käsittelyn sekä pakettien uudelleen lähetyksen.

FASP on käytössä laajasti media-alalla. Tuotantoyhtiöt siirtävät isoja videotiedostoja TV-yhtiöille käyttäen FASP-protokollaa. Tuotantoyhtiöt eivät usein tarjoa mitään muuta siirtokanavaa videotiedostoille kuin FASP:n, minkä takia sen käyttäminen on pakollista osana ympäristöä. Asperan eri siirto-sovellukset rakentuvat FASP-protokollan päälle, minkä takia tekniikan ymmärtäminen oli oleellinen osa ympäristön pystytystä. Tässä diplomityössä ei käydä kuitenkaan FASP-protokollan toimintaa tarkasti läpi sen ollessa lisensoitu ja suljettu protokolla, vaan työ keskittyy Asperan tuotteiden käyttöönottoon aineistonsiirtoympäristössä kokonaisuutena.

FASP-protokolla käyttää tiedonsiirtoprotokollana oletuksena UDP-protokollaa. UDP-protokollalla ei ole omaa sisäistä mekanismia virheenkäsittelylle, kuten TCP-pohjaisissa siirtoprotokollissa [3], eli se ei osaa itse lähettää siirrossa kadonneita paketteja uudestaan. FASP-protokolla korjaa tämän puutteen pitäen kirjaa lähetetyistä paketeista. Asiakasohjelma sekä palvelin keskustelevat keskenään erillisen kontrolliväylän kautta, jonka kautta asiakasohjelma voi pyytää palvelinta lähettämään korruptoituneet tai puuttuvat paketit uudestaan. FASP tarjoaa myös mahdollisuuden siirtää tiedostot HTTP-yhteyden yli, jos palvelimen ja käyttäjän välinen palomuuuri estää UDP-liikenteen. FASP osaa reagoida UDP-yhteyden estoon ja siirtyy automaattisesti käyttämään HTTP-protokollaa.

### 2.1.3 FASP- ja FTP-siirtojen vertailua

FASP-protokolla lupaa RTT:stä ja pakettien hukkumistodennäköisyydestä riippumattoman siirtonopeuden. Ainoina rajoitteina on siirtokanavan teoreettinen maksiminopeus, sekä FASP-palvelimen lisenssi.

Tätä vasten tein nopeusvertailua FASP:n ja FTP:n siirtonopeuksista. Simuloin verkon muutoksia Linux-käyttöjärjestelmiin löytyvällä netem-sovelluksella. Netem on verkkoemulaattori, joka hidastaa ohjelmallisesti verkkokortin lähettämiä paketteja. Hidastamalla pakettien lähetystä voidaan simuloida RTT:n kasvua verkossa. Samoin sovelluksella voidaan pudottaa paketteja halutulla todennäköisyydellä.

Kasvatin emulaattorilla yhteyden RTT:tä asteittain ja latasin 100 MB:n tiedoston samalla palvelimella olevalta FTP- ja FASP-palvelimelta omalle tietokoneelleni. Palvelin sijaitsi Suomessa ja on todellisuudessa se, joka asennettiin tämän projektin puitteissa myös lopulliseen tuotantokäyttöön. Käytössä ollut yhteys on Ethernet-tekniikkaan perustuva taloyhtiön laajakaista, jonka nimellinen kapasiteetti on 100/10 Mbit/s.

Ensin mitattiin ping-sovelluksella ilman hidastavaa emulaatiota palvelimen ja oman tietokoneen välinen RTT jonka arvoksi saatiin 2,5 ms. Tämän jälkeen kasvatettiin asteittain 50ms välein RTT:tä ja mittasin Wireshark-ohjelmalla ensimmäisestä paketista vii-

meiseen pakettiin kuluneen ajan. Jokainen mittaus toistettiin kolmesti sekä FASP- että FTP-protokollilla. Tuloksissa on esitetty näiden kolmen mittauksen keskiarvo. Tulokset on esitetty taulukoissa 1 ja 2.

**Taulukko 1: 100 MB tiedoston siirto FASP- ja FTP-protokollia käyttäen**

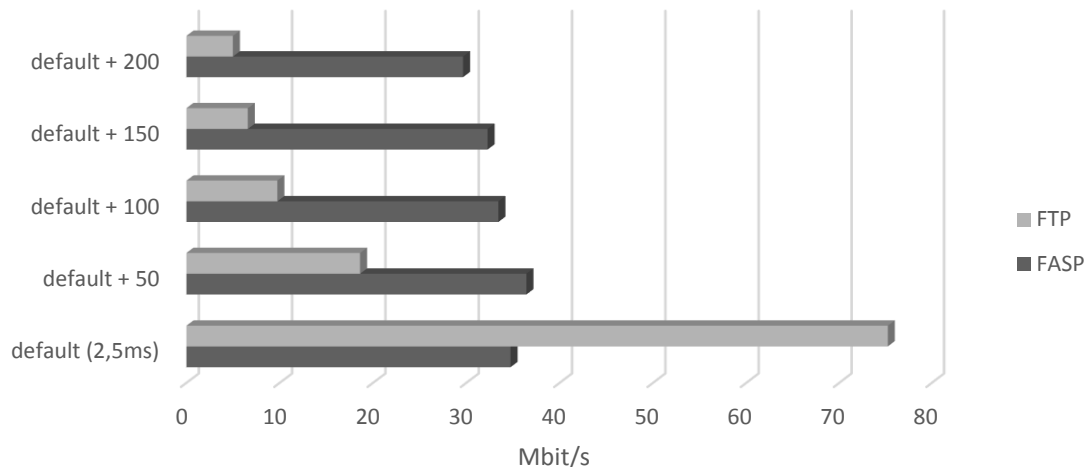
RTT (ms)	Aika FTP (s)	Aika FASP (s)
2,5	11,15	24,15
75	45,01	23,10
102,5	86,10	25,09
152,5	127,55	26,00
202,5	168,56	28,30

Siirtonopeus on laskettu kaavalla:

$$siirtonopeus = \frac{\text{paketin koko tavuina} * 8}{\text{siirtoon kulunut aika}}$$

**Taulukko 2: Siirtonopeudet FTP:tä ja FASPia käyttäen**

RTT (ms)	Siirtonopeus FTP (Mbit/s)	Siirtonopeus FASP (Mbit/s)
2,5	75,25	34,73
75	18,61	36,43
102,5	9,74	33,45
152,5	6,58	32,30
202,5	4,98	29,65



**Kuva 2: FASP ja FTP siirtonopeudet**

Tuloksista voidaan huomata, että FASP-protokollan suorituskyky pysyy lähes vakiona, kun taas FTP:n suorituskyky romahtaa, mitä kauempana kohde ja lähdepää ovat toisistaan. FASP:n suorituskykyä rajoittaa palvelinpäässä oleva lisenssi, joka rajoittaa yksittäisen siirron nopeudeksi 45 Mbit/s.

Mittaukset toteutettiin myös USA:ssa sijaitsevalle Asperan testipalvelimelle, jossa ei ole rajoituksia lisenssien osalta. Palvelimelta ladattiin 100 MB kokoinen tiedosto. Tulokset on esitetty taulukossa 3.

RTT (ms)	Aika FTP (s)	Aika FASP (s)
182	303,20	15,02

Kesimääräisiksi siirtonopeuksiksi saadaan:

FTP:  $(104857600 \cdot 8) \text{bit} / 303,20 \text{s} = 2,76 \text{Mbit/s}$

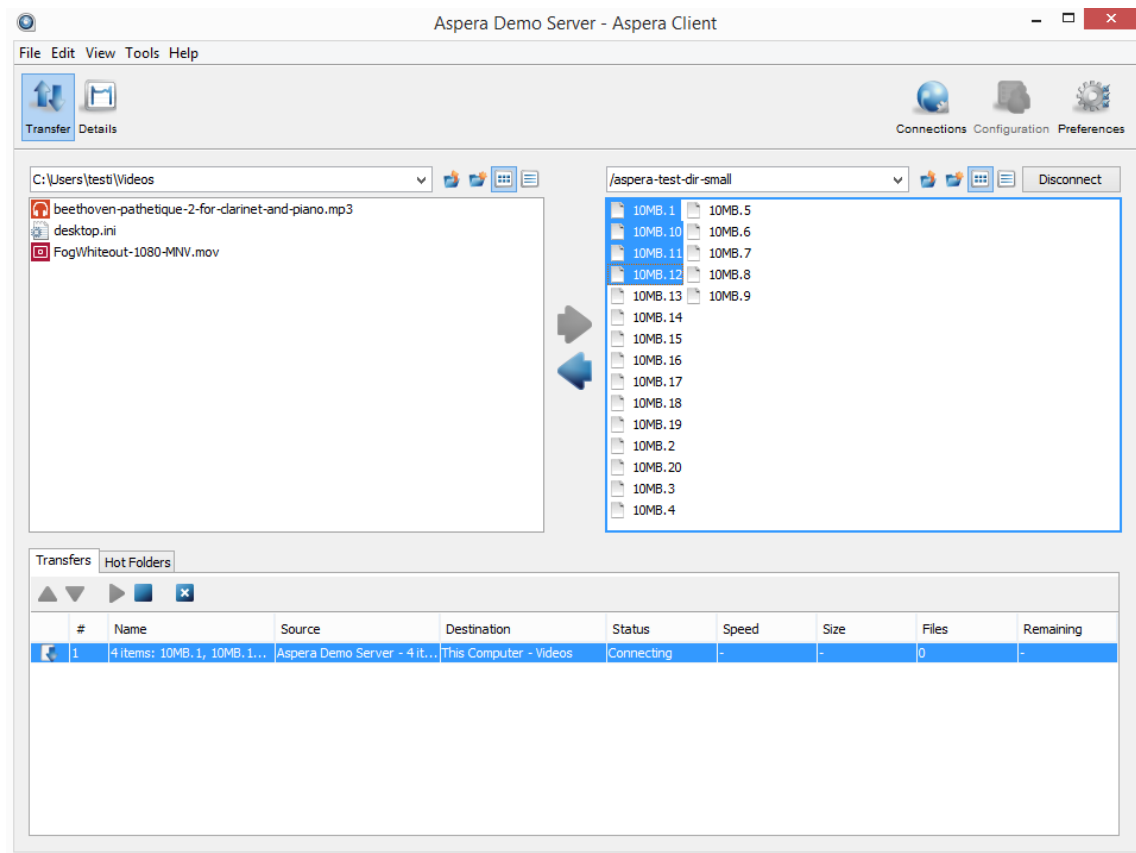
FASP:  $(104857600 \cdot 8) \text{bit} / 15,02 \text{s} = 55,85 \text{Mbit/s}$

Tässä ero kasvaa vielä suuremmaksi, kun simuloidussa mittauksessa.

## 2.2 Aspera Enterprise ja Connect Server

Aspera Enterprise Server on palvelinsovellus jonka avulla on mahdollista siirtää tiedostoja FASP-protokollan avulla. Enterprise Serverin asennus on pakollinen osa asennusta, jotta käyttäjät voivat siirtää tiedostoja erillisellä asiakasohjelmalla käyttäen FASP-protokollaa. Asiakasohjelma on saatavilla yleisimmille käyttöjärjestelmille. Asiakasohjelmasta on olemassa myös versiot iOS- ja Android-käyttöjärjestelmille, joiden avulla matkapuhelimella tai tabletilla kuvattua materiaalia on mahdollista ladata Enterprise Serverille ilman, että videota tai kuvaa tarvitsee siirtää ensin tietokoneelle. Asiakasohjelmat muistuttavat hyvin suuresti esimerkiksi FTP-asiakasohjelmia. Ensin käyttäjä ottaa yhteyden palvelimeen ja tämän jälkeen käyttäjän on mahdollista siirtää tiedostoja

palvelimen ja asiakasohjelman välillä. Tiedostojen poistaminen palvelimelta onnistuu myös asiakasohjelmaa käyttäen. Kuvassa 3 on Windowsin Aspera Client-asiakasohjelma.



**Kuva 3: Asperan Windows asiakasohjelma**

Asiakasohjelmasta on olemassa myös komentoriviltä käytettävät versiot, jonka avulla voidaan suorittaa tiedostonsiirtoja Linux-palvelimella ilman graafista käyttöliittymää.

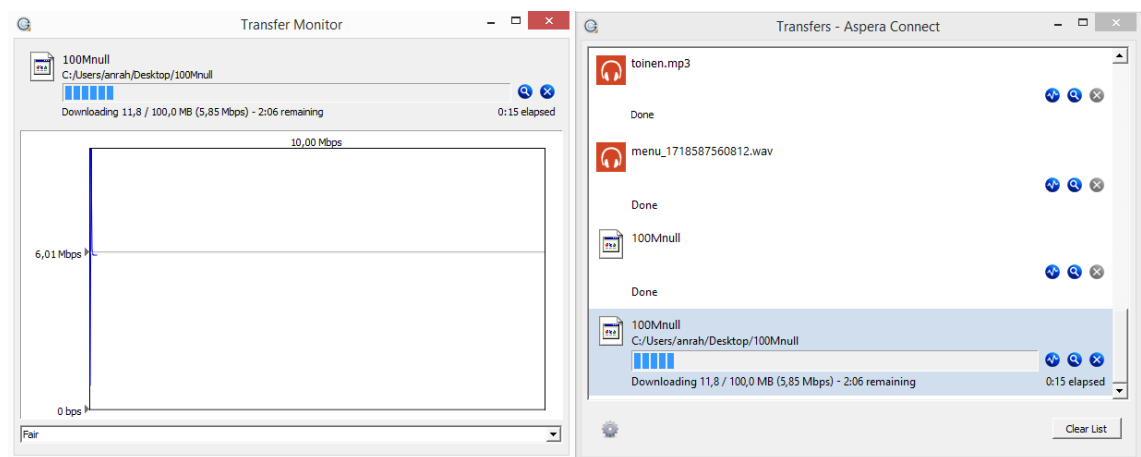
Kaikki asiakasohjelmat vaativat lisenssin, samoin kuin palvelinohjelmisto. Enterprise Serverin lisensointimalli perustuu siirtojen kaistanleveyteen. Lisenssi rajaa yksittäisen siirron kaistanleveyttä. Myöskin eri toiminnallisuuksia voidaan lisätä erilaisilla lisensseillä. Mobiilisiirrot esimerkiksi eivät kuulu osaksi peruslisenssiä, vaan mobiiliapplikaatioiden käyttö vaatii erillisen lisenssin palvelinpäässä.

Connect Server on laajennos Aspera Enterprise Server sovellukseen. Connect Serverin mukana tulee Web-sovellus, joka mahdollistaa käyttäjän ladata tiedostoja palvelimelle käyttäen Asperan Connect Client-selainlaajennosta. Käyttäjä pystyy lataamaan WWW-sivun kautta tiedostoja palvelimelle sekä palvelimelta samoin kuin poistamaan tiedostoja. WWW-sovelluksen käyttöliittymä on esitetty kuvassa 4. Aspera Connect Server on hyödyllinen sovellus silloin kun on tarvetta tarjota useille eri käyttäjille mahdollisuus ladata tiedostoja hyödyntäen FASP-protokollaa.



#### Kuva 4: Aspera Connect Server käyttöliittymä

Enterprise Serverin tapauksessa siirtoihin vaaditaan myös asiakaspäähän oma lisenssi, mutta Connect Serverin avulla tästä vaatimuksesta päästään eroon, sillä lisenssi on tällöin palveluntarjoajan vastuulla ja on osa palvelimen lisenssiä. MTV:n tapauksessa materiaalin toimittajia on useita kymmeniä, joten Connect Server lisenssi on pakollinen vaatimus. Toimijat voivat olla pieniä mainostoimistoja, joilla ei ole resursseja hankkia vaadittavia lisenssejä. Koska tavoitteena on saada kaikki materiaalitoimittajat käyttämään Aspera-siirtoja FTP-siirtojen sijaan, Connect Server oli ainoa vaihtoehto. Connect Serveriä käytettäessä käyttäjät lataavat tiedostoja palvelimelle tai palvelimelta ilmaisen Connect Client selainlaajennoksen avulla. Connect Client käynnistyy, kun käyttäjä valitsee Connect Serverin käyttöliittymästä haluamansa tiedoston ja valitsee haluavansa ladata sen. Vastaavasti kun käyttäjä lähettää tiedostoja palvelimelle, käyttäjälle aukeaa ikkuna, josta hän valitsee haluamansa tiedoston ja Connect Client käynnistää varsinaisen siirron. Kuvassa 5 on esitetty Connect Clientin näkymä kun tiedostoja siirretään palvelimelle.



#### Kuva 5: Aspera Connect Client

Connect Clientin siirrossa käyttäjällä on mahdollisuus valita haluamansa siirtonopeus liikuttamalla kuvassa 3 6,01 Mbps kohdalla olevaa liukuria ylös- tai alaspäin. Maksimissaan käyttäjä voi valita siirtonopeudeksi palvelimen lisenssin salliman siirtonopeuden.

Connect Server tarjoaa myös rajapinnan, jota on mahdollista kutsua ohjelmallisesti. Rajapinnan kautta on mahdollista siirtää tiedostoja palvelimen ja asiakasohjelman välillä sekä poistaa tiedostoja palvelimelta. Rajapinta mahdollistaa omien web-sovellusten kehittämisen, jotka käyttävät tiedostonsiirrossa hyväksi Asperan Connect Clientia. Esimerkiksi voidaan toteuttaa erillinen HTML-sivu, jossa täytetään tarvittava metatietolomake toimitettavia tiedostoja varten ja varsinainen tiedostojen siirto tapahtuu käyttäen Aspera Connect Clientia perinteisen HTTP-siirron sijaan.

## 2.3 Aspera Orchestrator

Orkestroinnilla tarkoitetaan ennalta määrättyjen tehtävien automaattista toteuttamista, jotka käynnistyvät jonkin tietyn tapahtuman tai syötteen seurauksena. Orkestrointisovelluksissa rakennetaan työnkulkua liittämällä eri toimintoja toisiinsa. Tehtävät ovat pieniä ja ennalta määriteltyjä, jolloin monimutkainen toiminnallisuus saadaan toteutettua ketjuttamalla tarpeeksi pieniksi palasiksi pilkottuja toiminnallisuuksia toisiinsa. Yleisesti myös toiminnallisuus voi muuttua riippuen syötteen arvoista, eli toiminnallisuutta parametrisoidaan kesken suorituksen.

Aspera Orchestrator on Asperasoftin kehittämä orkestrointisovellus, jonka avulla luodaan työnkulkua ja jotka laukaistaan käyntiin tietyn ennalta määritellyn tapahtuman seurauksena. Työnkulku suorittaa määritellyt tehtävät ja palauttaa tiedon suorituksen onnistumisesta tai epäonnistumisesta. Suorituksen epäonnistuessa sovellus palauttaa epäonnistuneen tehtävän tiedot ja mahdollisen virheilmoituksen.

Markkinoilla on tarjolla muitakin orkestrointisovelluksia, mutta ne eivät tarjoa valmiita toteutuksia, jolla Asperan muut tuotteet on mahdollista integroida osaksi työnkulkua. Samoin Asperan Orchestrator tarjoaa valmiit toteutukset eri transkoodausjärjestelmiin, mikä oli oleellinen osa vaatimusta. Suuri valmiiden integraatiototeutusten määrä MTV:llä käytössä oleviin järjestelmiin oli ratkaiseva tekijä, miksi MTV halusi ottaa Orchestrator-sovelluksen osaksi aineistosiirtojärjestelmää.

Sovelluksessa on valmiina toteutettuna seuraavia toimintoja:

- Laukaisimet (trigger) laukaisimet käynnistävät työnkulkujen suorittamisen. Laukaisimia on muutamaa eri tyyppiä toteutettuna valmiina:
  - Ajastettu – Työnkulku suoritetaan tietyssä ajanhetkenä joko kerran tai toistuvasti.
  - Uuden tiedoston tai hakemiston tarkkailija – Työnkulku suoritetaan kun tarkasteltavaan hakemistoon ilmestyy uusi tiedosto tai hakemisto. Tarkkailijaan voidaan määrittää erilaisia suodattimia, jonka avulla voidaan jättää huomiotta tiettyyn säännölliseen lausekkeeseen täsmäävät tiedostot tai hakemistot. Nämä laukaisimet palauttavat ulostulossaan löydetty tiedostot tai hakemistot, jolloin niitä voidaan käyttää parametreina seuraavassa lohkoissa.



- FTP-kotihakemiston tarkkailija – Laukaisin tarkastelee määritellyn FTP-palvelimen ja käyttäjän kotihakemistoa ja palauttaa parametrina saapuneet tiedostot.
  - Asperan sovelluksien tilasiirtymät – Laukaisin tarkastelee esimerkiksi Connect Serverille tapahtuvan siirron valmistumista ja palauttaa parametrina tapahtuman tiedot.
  - Tietokantatapahtuma – Laukaisin tarkastelee tietokantakyselyn kautta tietokannassa ilmenevää tapahtumaa ja palauttaa parametrina käyttäjän määrittelemät tiedot. Käyttäjän on mahdollista prosessoida tietokantakyselyn tulosta ja palauttaa laukaisimesta halutut parametrit
  - Sähköpostilaatikon tarkkailija – Laukaisiin tarkkailee määriteltyyn sähköpostilaatikkoon saapuvia viestejä ja palauttaa sähköpostin otsikkotiedot, rungon sekä mahdolliset liitetiedostot
  - AMQP-jono – Laukaisin tarkkailee määriteltyä AMQP-jonoa ja palauttaa parametrina jonoon saapuneen viestin.
  - SOAP-kuuntelija – Laukaisin kuuntelee Orchestrator-sovelluksen SOAP-porttia ja laukaisee määritellyn työnkulun. Orchestrator palauttaa SOAP-kutsun paluuviestinä käyttäjän määrittelemän vastauksen.
  - Itse luotu – Käyttäjän on itse mahdollista luoda Ruby-ohjelmointikieltä käyttämällä oma laukaisin.
- Tiedostonkäsittelijät
- Tiedoston kopiointi tai siirto paikallisesti
  - FTP-siirrot
  - SCP-siirrot
  - Aspera-siirrot
  - Tiedoston metatietojen haku - esimerkiksi audio- tai video-tiedoston laadutiedot
  - Arkistointi – tiedostojen pakkaaminen ja automaattinen siivous esimerkiksi aikaleiman perusteella
  - XSLT-muunnos – XSLT:n avulla on mahdollista muuntaa XML-pohjaisia dokumentteja toiseen XML-malliin.
  - Laadunvarmistus – eri laadunvarmistustyökaluilla voidaan todentaa esimerkiksi audiotiedoston täyttävän tietyt laatuvedellytykset
- Tiedostonkäsittelijät tarvitsevat parametreina käsiteltävän tiedoston, sekä kopiointien tai siirtojen tapauksessa kohdepalvelimen sekä kohdepolun. Suoritukset palauttavat tapahtuman tuloksen.
- Integraattorit
- SOAP-kysely – Orchestrator voi luoda SOAP-kutsuja ulkoisiin järjestelmiin

- REST-kyselyt – Orchestrator voi luoda REST-metodeilla kutsuja ulkoi-  
siin järjestelmiin
  - AMQP-viestien lähetys
  - Ulkoinen suoritus – Orchestrator voi SSH:n yli käydä suorittamassa ko-  
mentoja ulkoisella palvelimella
  - Tietokantakysely – Orchestrator voi suorittaa tietokantakyselyjä ulkoisiin  
tietokantoihin
  - Transkoodaus – Orchestrator voi kutsua yleisimpiä transkoodaus-  
järjestelmiä rajapintojen kautta.
  - Sähköposti-ilmoitukset
  - Käyttäjän tehtävät – Orchestratoriin on mahdollista luoda tehtäviä (task),  
joiden perusteella käyttäjälle voidaan antaa mahdollisuus valita seuraava  
suoritettava operaatio.
- Sisäiset toiminnot
- Talletetut parametrit – Työnkulkuun liittyvät parametrit voidaan tallettaa  
omaan lohkoonsa, jolloin niitä voidaan helposti muokata.
  - Resurssienhallinta – Työnkuluille voidaan luoda resursseja. Esimeriksi  
jos käytössä on kaksi transkoodausinstanssia, niin resurssienhallinnan  
avulla voidaan luoda jono käsiteltävistä tiedostoista. Jonosta pääsee läpi  
seuraava tiedosto vasta kun edellisen tiedoston transkoodaus on suoritet-  
tu loppuun. Käyttäjälle annetaan myös mahdollisuus siirtää jonossa ole-  
via elementtejä ylös- tai alaspäin, jolloin kiireelliset elementit voidaan  
käsitellä ensin.

Jokainen toiminto vaatii joko kiinteästi määritellyt tai suorituksen aikana määräytyvät parametrit. Kiinteästi määritellyt parametrit ovat yleisesti tiettyyn työnkulkuun liittyvät laadunvarmistuskriteerit sekä tiedostojen lähde- tai kohdehakemistot. Suorituksen aika-  
na määräytyvät parametrit muodostuvat työnkulun suorituksen mukaan ja niiden perus-  
teella voidaan tehdä ehdollisia toimenpiteitä työnkulun sisällä.

Työnkulun on mahdollista kutsua toista työnkulkua. Kutsuminen voi tapahtua joko ali-  
työnkulkuna, jolloin kutsuva työnkulku odottaa alityönkulun valmistumista ja jatkaa  
suoritustaan sen jälkeen. Toinen vaihtoehto on laukaista toinen, täysin irrallinen työn-  
kulku. Tällöin kutsuva työnkulku ei odota laukaistun työnkulun valmistumista, vaan  
jatkaa normaalisti eteenpäin.

Valmiiksi toteutettujen rajapintojen suuri määrä jo käytössä oleviin järjestelmiin oli  
pääasiallinen syy miksi MTV päätyi hankkimaan Orchestratorin osaksi aineistonsiirto-  
ympäristöä. Integraatiot Connect Serveriin ja Faspexiin mahdollistavat työnkulkujen  
luomisen siten, että käyttäjän ei tarvitse käsin hoitaa tiedoston siirtoa käyttäen Connect  
Serveriä, vaan kaikki toiminnallisuus voidaan siirtää suoraan Orchestratoriin.

Tarkemmin sovelluksen toimintaa käydään läpi luvussa 5 erilaisten toteutettujen työnkulkujen avulla.

## 2.4 Aspera Faspex

Siinä missä Connect Server sekä Enterprise Server on tarkoitettu pääasiallisesti organisaatioiden välisiin tiedostonsiirtoihin, on Faspex tarkoitettu yksittäisten käyttäjien tai organisaation sisäisten tiimien tai käyttäjäryhmien välisiin tiedostonsiirtoihin. Faspex käyttää edelleen tiedoston siirtomekanismina Enterprise Serveriä, mutta tuo tähän päälle lisää toiminnallisuutta.

Yleinen käyttötapaus vanhassa ympäristössä oli sellainen, missä käyttäjä haluaa lähettää toiselle käyttäjälle tai käyttäjäryhmälle suuren mediatiedoston. Käyttäjällä oli mahdollisuus vanhassa ympäristössä jakaa tiedosto joko FTP-palvelimen tai verkkolevyn kautta. Ongelmaksi näissä tavoissa muodostui se, että FTP ei tarjoa mahdollisuutta ilmoitta esimerkiksi sähköpostilla suoraan käyttäjille uudesta tiedostosta. Samoin jos mediatiedosto on esimerkiksi HD-laatuinen elokuva, jonka koko voi olla useita kymmeniä gigatavuja, kestää tiedoston lataaminen kauan sekä palvelimelle että toisten käyttäjien ladatessa se palvelimelta.

Jaetun verkkolevyn ongelmaksi muodostuu se, jos osa vastaanottajista sijaitsee organisaation ulkopuolella, ei heillä ole pääsyä verkkolevylle ilman VPN-yhteyttä.

Usein on myös tarve liittää tiedoston mukaan metatietoja, jotka kertovat mihin kyseinen tiedosto liittyy. Tämä tarvitsee tällöin tehdä erillisen tekstitiedoston avulla.

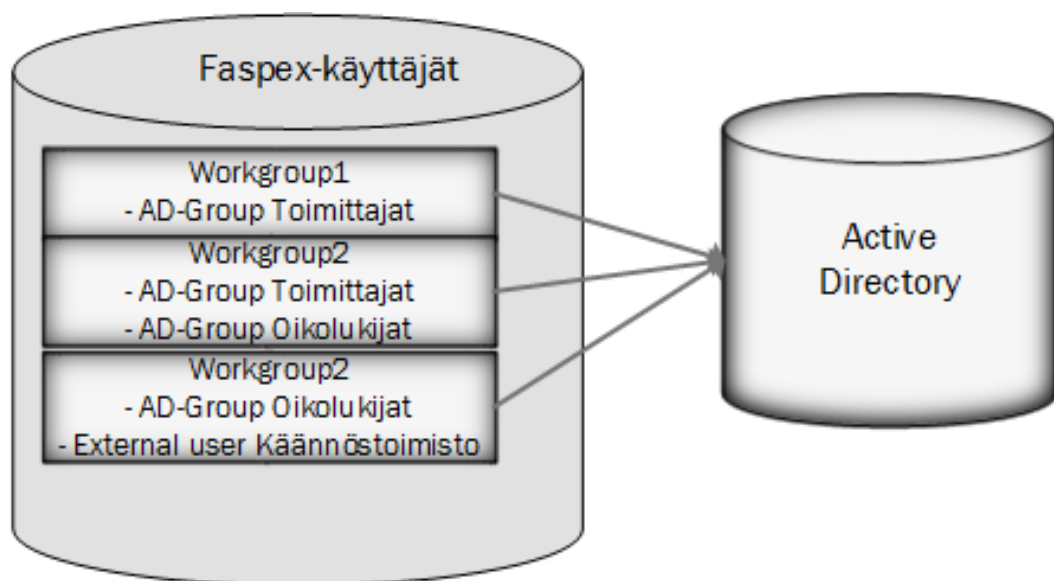
Faspexin toiminta ja siirrot perustuvat Faspex-paketteihin. Käyttäjä luo joko WWW-sivun kautta tai mobiilisovelluksella Faspex-paketin, joka voi sisältää yhden tai useamman tiedoston. Käyttäjä lähettää paketin haluamalleen vastaanottajalle tai vastaanottajajoukolle. Kun paketti on ladattu, vastaanottaja tai vastaanottajat saavat sähköposti-ilmoituksen saapuneesta paketista sekä latauslinkin, josta paketin voi hakea. Faspexissa on ominaisuus, jonka avulla käyttäjä pystyy lähettämään organisaation ulkopuoliselle käyttäjälle kertakäyttöisen latauslinkin.

Aikaisemmin MTV:llä ei ollut käytössä Faspexia ja tiedostojen siirrot organisaation sisällä hoidettiin perinteisesti verkkolevyjen avulla. Organisaation ulkopuolelle tarkoitettut tiedostot jaettiin FTP:n avulla. Metatietojen puuttumisen ohella ongelmaksi muodostui tiedostojen poistaminen sen jälkeen, kun materiaali oli toimitettu. Vanhoja tiedostoja saattoi olla jopa satojen gigatavujen edestä, jotka käyttivät turhaan levytilaa. Faspexissa on ominaisuus, joka mahdollistaa pakettien automaattisen poistamisen ajastetusti tai heti sen jälkeen kun kaikki vastaanottajat ovat ladanneet paketin itselleen. Tämä vähentää ympäristössä tarvittavaa ylläpidon määrää, kun paketit poistuvat automatisoidusti palvelimelta.

Faspexissa on sisäänrakennettu käyttäjänhallinta, mutta Faspex on mahdollista liittää myös organisaation LDAP:iin [4] tai Microsoftin Active Directoryyn [5], jolloin käyttäjät autentikoituvat sovellukseen käyttäen organisaation käyttäjätunnuksia.

Faspexin käyttäjähallinta perustuu työryhmiin. Työryhmän sisällä käyttäjät voivat lähettää toisilleen vapaasti paketteja. Työryhmät voivat muodostua automaattisesti LDAP:in OU-haaroista, tai niitä on mahdollisuus luoda käsin Faspexiin. Yhteen työryhmään voi kuulua monia eri AD-ryhmiä.

Oletuksena käyttäjän ei ole mahdollista lähettää paketteja oman työryhmänsä ulkopuolisille käyttäjille. Asetukset ovat kuitenkin muokattavissa, jolloin voidaan sallia työryhmän jäsenelle oikeus kutsua ulkopuolisia käyttäjiä osaksi työryhmää. Työryhmä on perustoiminnallisuus, jossa jokainen työryhmän jäsen näkee kaikki työryhmään toimitetut paketit. Työryhmän jäsenillä voi olla eri oikeuksia. Peruskäyttäjällä ei ole muita oikeuksia kuin toimittaa tai ladata paketteja. Työryhmän pääkäyttäjällä on mahdollisuus poistaa paketteja sekä lisätä käyttäjiä työryhmään. Käyttäjähallinnan periaate on esitetty kuvassa 6.



**Kuva 6: Faspex-käyttäjähallinta**

Dropbox-toiminnallisuus on muuten vastaava kuin työryhmä, mutta ladattaessa dropboxiin tiedostoja on käyttäjän täytettävä metatietolomake (kuva 7), jotka näkyvät paketin tiedoissa muille käyttäjille kun siirto on valmis (kuva 8).

## Send to Dropbox: \*Tiketti

**Instructions**  
Lisää tikettinumero

[Hide Cc](#)

Cc

Upload: ☐ anrah, +

Download: ☒ anrah, +

Notifications are sent after first upload or download

Title: Testipaketti

Tikettinumero: UUUXXX-553


Note: optional  
Tässä tiedostoja.

Encryption: ☐ Use encryption-at-rest

Contents:

OR

Drop Files and Folders Here



beethoven-pathetique-2-for-... C:\Users\anrah\Desktop\beethoven-pathetique-2-fo...

**Kuva 7: Faspex dropbox - Lähettäminen palvelimelle**

Package - Uusi paketti [Return to List](#) [Forward](#)

From: anrah  
To: \*Tiketti  
Cc (on download): anrah,  
Date sent: 12/16/14 02:49 PM  
Tikettinumero: DACEOPS-666  
Note: Testipaketti

Browse and Download Contents [Download Entire Package](#)

Name	Size
beethoven-pathetique-2-for-clarinet-and-piano.mp3.aspera-env	4.3 MB

Download selected [Select All](#) [None](#)

**Package Details**  
Status: Complete  
Size: 4.3 MB  
Files: 1  
Upload status:  
Elapsed: less than 5 seconds  
Average rate: 5.5 MBps  
Uploaded from: 10.31.72.185  
Full Package Downloads: 0  
Partial Package Downloads: 0  
Active Downloads: 0

**Kuva 8: Faspex dropbox - Paketin lataaminen palvelimelta**

Faspex-paketit on mahdollista salata palvelimella. Ilman salausta paketit näkyvät palvelimen levyllä normaaleina hakemistoina, joiden sisällä on pakettiin kuuluvat tiedostot. Jos toimitettava materiaali on luottamuksellista, voi lähettäjä määrittää paketille salausavaimen, jota ilman pakettia ei pysty avaamaan tai lataamaan. Salaus voi toimia kahdella eri tavalla: Joko siten, että käyttäjän on syötettävä salausavain vain tiedoston lataamisen yhteydessä, jolloin paketti siirtyy salattuna käyttäjän omalle tietokoneelle, mutta pakettia ei ole enää erikseen salattu käyttäjän kovalevyllä. Toinen vaihtoehto on, että

käyttäjän on syötettävä salausavain joka kerta kun hän haluaa avata paketissa olevia tiedostoja. Tällöin Connect Client kysyy salausavainta paketille joka kerta kun käyttäjä yrittää avata pakettia.

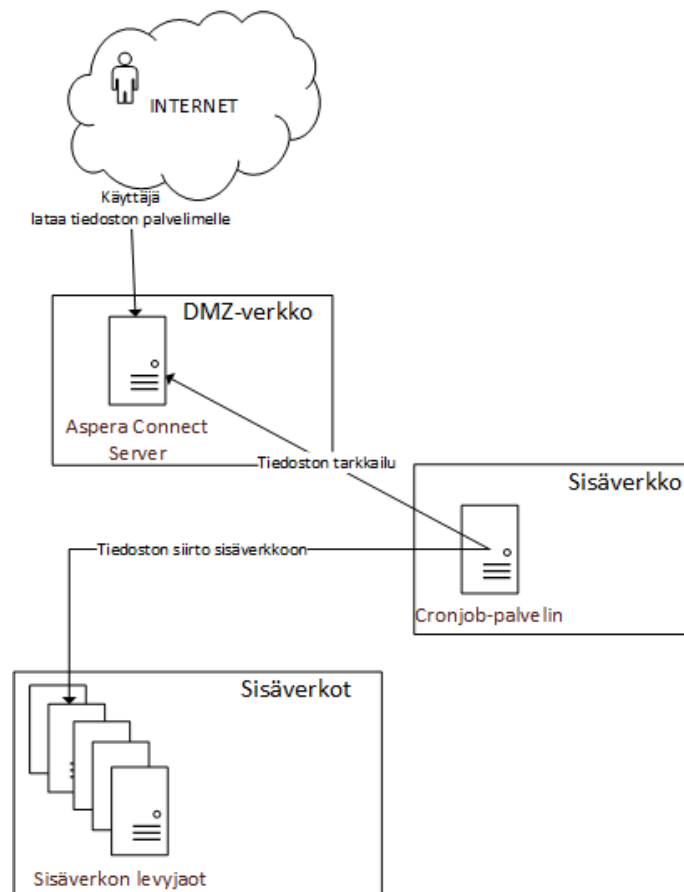
## **2.5 Aspera Console**

Aspera Console on tarkoitettu Connect Serverin hallintaan ja monitorointiin. Consolen graafisen käyttöliittymän kautta pystyy tarkastelemaan käynnissä olevia siirtoja, etsimään virhetilanteissa lokeja sekä hallitsemaan Connect Serverin asetuksia. Ilman Consolea Connect Serverin hallinta tapahtuu suoraan komentoriviltä. Consolen kautta on myös mahdollisuus luoda ajastettuja siirtoja käyttäen Connect Serveriä. Samoin Console tarjoaa laajat raportointimahdollisuudet kaikista Connect Serverin siirtojen osalta. Näitä raportteja voidaan käyttää tarvittaessa vianselvityksessä, kun etsitään tietoa siitä onko joku tiedosto saapunut palvelimelle.

### 3 LÄHTÖ- JA TAVOITETILANNE

MTV:llä oli projektin alkaessa käytössä Asperan tuotteista ainoastaan Asperan Connect Server. Käyttäjinä järjestelmällä olivat tuotantoyhtiöt, mainostoimistot sekä MTV:n omat toimittajat. Jokaisella toimijalla oli oma tunnus joko Connect Serverin käyttöön tai vaihtoehtoisesti FTP-palvelimelle, joka sijaitsi samalla palvelimella kuin Connect Server. Kaikki toimitukset tapahtuivat tälle DMZ-verkossa sijaitsevalle tiedostonsiirto-palvelimelle.

DMZ-verkon palvelimelle saapuneet tiedostot käsiteltiin sisäverkossa olevan Cronjob-palvelimen avulla automaattisesti. Vaihtoehtoisesti käyttäjät kävivät käsin tarkastamassa oliko uusia tiedostoja saapunut. Ympäristön periaatekuva on esitetty kuvassa 9.



**Kuva 9: Vanhan ympäristön periaatekuva**

Tiedostoja käsittelevä Cronjob-palvelin oli Linux-palvelin, minne oli tallennettu useita crontab-sovelluksen [6] avulla ajastettuja Bash-skriptejä. Crontab-sovelluksen suorittamat skriptit suoritettiin minuutin, puolen tunnin tai tunnin välein riippuen tiedostojen kiireellisyydestä. Crontab-palvelimella oli käytössä sama levyjako kuin DMZ-verkossa

olevalla tiedostonsiirtopalvelimella. Skriptit tutkivat oliko skriptin vastuualueisiin kuuluihin hakemistoihin ilmestynyt uusia tiedostoja. Skriptien vastuualueisiin kuuluvat hakemistot vastasivat siirtokäyttäjien kotihakemistoja. Jos uusia tiedostoja oli saapunut, skriptit siirsivät ne pois DMZ-verkosta määritellyyn kohdehakemistoon verkkolevyille, joka oli käytössä ainoastaan sisäverkon palvelimilla. Siirtojen valmistuttua käyttäjille lähetettiin viesti, että uusi tiedosto on saapunut. Vastaavasti virhetilanteissa käyttäjiä informoitiin virhetilanteesta. Skriptit kirjoittivat myös lokitiedostoa palvelimelle, josta tarvittaessa voitiin käydä tarkistamassa oliko tietty tiedosto saapunut.

Varsinaisen ympäristön ulkopuolella oli myös erilaisia tapoja toimittaa materiaalia MTV:n käyttöön. Käytössä oli muun muassa sähköpostilaatikoita, joista liitetiedostot talletettiin käsin tarvittaviin paikkoihin. Ympäristössä oli myös useita verkkolevyjä, sekä osalla palvelimista suuret paikalliset levyt, joita jaettiin eri käyttötapauksia varten.

### 3.1 Lähtötilanteen ongelmat

MTV tavoitteena oli yhtenäistää materiaalinkäsittely yhteen paikkaan, jolloin kaikki sisään- tai ulospäin liikkuva materiaali käsiteltäisiin samassa paikassa. Käytössä olleen ympäristön rakenne ei tukenut tätä tavoitetta. Vanhan järjestelmän tiedostonkäsittelijät DMZ-verkon ja sisäverkon välillä perustuivat kokonaisuudessaan Bash-skripteihin, jotka oli toteutettu MTV:n sisällä. Niiden ylläpito oli myös MTV:n vastuulla. Tavoitteen mukaisesti kaikki käsittely tulisi tapahtua yhdessä paikassa, jolloin uusien käsittelijöiden toteuttaminen olisi onnistunut muokkaamalla olemassa olevien käsittelijöiden kohde- ja lähdehakemistoja. Bash-skriptien luomista uusiin tarpeisiin ei kuitenkaan nähty MTV:n sisällä järkeväksi tavaksi toteuttaa järjestelmää. Orchestrator-sovelluksen tuomat uudet ominaisuudet kuten sähköpostilaatikoiden tarkkailu, Faspex-laatikoiden tarkkailu, tiedostojen automaattinen lähetys Asperan Connect Serverin avulla ulospäin sekä integraatiomahdollisuudet muihin järjestelmiin, kuten transkoodereihin toivat mukanaan lisähyötyjä, jotka haluttiin ottaa käyttöön. Kokonaan uuden ympäristön rakentaminen sekä Orchestrator-sovelluksen käyttöönotto tulisi kannattavammaksi, kuin lähteä toteuttamaan uusia käsittelijöitä olemassa olevien skriptien pohjalta.

Vanha ympäristö oli ollut pääosin ainoastaan ohjelma- ja mainosmateriaalin käsittelijöiden käytössä. Tavoitteena oli saattaa ympäristö myös muiden MTV:n yksiköiden käyttöön siten, että ne voisivat siirtää omia aineistonsiirtoon liittyviä toteutuksiaan uuden ympäristön vastuulle.

Orchestrator-sovelluksen myötä ympäristöön olisi mahdollista toteuttaa sellaisia työnkulkuja, jotka eivät olleet mahdollisia vanhojen Bash-skriptien avulla. Esimerkkinä saapuvalla materiaalille suoritettavat automaattiset laadunvarmistustehtävät. Uusien työnkulkujen tarkoituksena oli vähentää käsin tapahtuvaa laadunvarmistusta siten, että jokaiselle saapuvalla mediatiedostolle suoritetaan yksinkertainen laadunvarmistus *Mediainfo*-ohjelman avulla. Mediainfon avulla saadaan tarkastettua saapuvan materiaalin laatutiedot. Jos ne eivät täytä vaatimuksia, voidaan tiedosto hylätä jo alkuvaiheessa,



eikä loppukäyttäjän tarvitse suorittaa näitä tarkistuksia käsin. Jos tiedosto on korruptoitunut jo siirtovaiheessa, tämä saadaan myös tarkastettua helposti mediainfon avulla. Korruptoituneessa tiedostossa metatiedot ovat yleensä rikki, eikä tiedostoa pysty avaamaan mediainfon avulla. Näissä tapauksissa voidaan informoida lähettäjästä automaattisesti, joka toimittaa materiaalin uudelleen.

Lisäksi vanha järjestelmä oli ollut käytössä vuodesta 2009. Käyttöjärjestelmän iän ja kriittisyyden takia järjestelmä olisi pitänyt asentaa uusiksi uusille palvelimelle. Uudistuksen yhteydessä tavoitteena oli myös vähentää ympäristössä levyjakojen määrää, sekä korvata palvelinten paikalliset levyt yhdellä isolla levyjaolla, jota jaettaisiin ympäristössä käyttötapauksen mukaan. Koska MTV:llä oli käytössä suorituskykyinen levypalvelin, pyrittiin sen käyttöä lisäämään osana ympäristön kehitystä. Tavoitteena oli päästä eroon palvelinten fyysisistä levyistä ja siirtää kaikki tiedostot keskitetyille levyjärjestelmälle. Siirtämällä DMZ-verkkoon ja sisäverkkoon näkyvät yhteiset levyalueet yhden levyjärjestelmän vastuulle, päästäisiin eroon paikallisista levyjaoista, joiden vikatilanteiden selvittäminen ei ole yhtä helppoa kuin keskitetystä paikasta jaeltu levytila. Myös levyjakojen käyttäjäoikeushallinta on helpompaa yhdestä paikasta. Levyjärjestelmän kautta jaettuna pystyttäisiin jakamaan samaa levyjakoa sekä käyttäjien työpöydälle, että aineistonsiirtoympäristön palvelimille ja hoitamaan käyttäjähallinta suoraan MTV:n sisäisen Active Directoryn kautta.

Ylläpitoon liittyvänä ongelmana nähtiin, että cronjob-palvelimen lokitiedostoihin ei ollut pääsyä kuin järjestelmäylläpitäjillä. Osa materiaalista on erittäin aikakriittistä, kuten esimerkiksi uutisiin tarkoitettu materiaali. Tämä tarkoitti sitä, että materiaalinkäsittelijän pitäisi päästä itse tutkimaan tarvittaessa lokitiedostoja, jotta hän voi varmistua siitä, että haluttu materiaali on todella toimitettu. Harjaantumattomalle Linux-käyttäjälle lokitiedoston selaaminen Linuxin komentorivityökaluilla voi olla hankalaa. Kaikista vikatilanteista ei välttämättä saatu ajallaan virheilmoitusta. Esimerkiksi levyjaon katoaminen palvelimelta huomattiin yleensä vasta kun levyjaolle oltiin kirjoittamassa. Jos kyseessä oli hieman vähemmän käytetty levyjako, saattoi vikatilanne olla päällä useita päiviä. Valvontaa oli siis kehitettävä samalla kun järjestelmää päivitetään.

## 3.2 Tavoitetila

Tavoitteena oli siirtää kaiken sisään- ja ulospäin liikkuvan materiaalin käsittely uuden ympäristön vastuulle. Orchestrator-sovelluksen avulla voidaan luoda sellaisia työnkuluja, jotka ovat helposti monistettavissa siten, että niitä voidaan vain parametrisoida eri käyttötarkoituksiin. Näin saadaan toteutettua yhdellä toteutuksella useimpiin eri käyttötarkoituksiin sopiva työnkulku, jossa on kaikki tarvittavat loikitus- ja valvontatarpeet toteutettuna. Samoin ympäristön valvonta ja lokitiedostojen käsittely pyrittiin keskittämään Orchestrator-sovellukseen, joka valvoo automaattisesti työnkulkujen suoritusta ja osaa hälyttää tarpeen mukaan oikeita henkilöitä.

Orchestrator-sovelluksen tuomien lisäominaisuuksien kautta oli myös tarkoitus korvata ja automatisoida materiaalinkäsittelyssä myöhemmässä vaiheessa tapahtuvia laadunvarmistukseen liittyviä toimenpiteitä. Esimerkiksi *mediainfo*-laajennoksen avulla pystytään tarkastelemaan saapuvan materiaalin laatutietoja. Myös MTV:n käyttämään transkoodausjärjestelmään, jolla saapuva materiaali transkoodataan TV-lähetykseen sopivaksi, oli olemassa valmis lisäosa Orchestrator-sovelluksessa. Aikaisemmin tätä tarkoitusta varten oli erillinen järjestelmä, joka suoritti saapuvien tiedostojen siirtämisen transkoodaukseen. Käyttäjällä ei kuitenkaan ollut mitään mahdollisuutta valita tiedostojen transkoodausjärjestystä. Uuteen järjestelmään oli tarkoitus toteuttaa tätä varten uusi työnkulku, jonka avulla materiaalinkäsittelijä pystyi priorisoimaan saapuvaa materiaalia.

Vanhan ympäristön puutteet saapuvan materiaalin lokituksessa ja läpinäkyvyydessä oli tarkoitus saattaa kuntoon Orchestrator-sovelluksen tarjoaminen ominaisuuksien kautta. Sovelluksen käyttöliittymän kautta materiaalinkäsittelijä pystyy tarkkailemaan milloin materiaali on saapunut ja missä vaiheessa työnkulkua se on menossa. Samoin jos työnkulussa on tapahtunut virhe, niin sähköposti-ilmoituksen perusteella materiaalinkäsittelijällä on mahdollisuus saada paremmin tietoonsa miksi tiedosto ei ole läpäissyt työnkulkua. Nämä ominaisuudet puuttuivat täysin vanhasta ympäristöstä. Aikaisemman ympäristön tapauksessa ilmoitukset lähtivät ainoastaan kun tiedostoa ei saatu siirrettyä DMZ-verkosta sisäverkkoon tai mahdollisesti siitä, jos Bash-skripti oli kaatunut kesken suorituksen. Käyttäjillä ei kuitenkaan ollut mahdollisuutta käynnistää siirtoa uudelleen itse.

Faspex-sovelluksen käyttöönoton myötä käyttäjille oli tarkoitus tarjota helppokäyttöinen, käyttäjältä käyttäjälle tarkoitettu tiedostonsiirtomekanismi. Faspex-sovelluksen dropbox-toiminnallisuus sekä työryhmät vastaisivat tarpeeseen, jossa tarvitsee siirtää yksittäinen tai useampi tiedosto talon sisällä toiselle käyttäjälle. Faspex-sovelluksen myötä tavoitteena oli myös päästä eroon ylimääräisistä FTP-tunnuksista. Vanhassa järjestelmässä yksittäistäkin toimitusta varten MTV:n ulkopuolelle vaadittiin käytännössä uuden FTP-tunnuksen luonti, jos kohde ei ollut käyttänyt mitään MTV:n järjestelmiä aikaisemmin. Nämä FTP-tunnukset jäivät helposti siivoamatta käytön jälkeen. Faspexin ominaisuus, jossa käyttäjä pystyy luomaan organisaation ulkopuoliselle toimijalle kertakäyttöisen latauslinkin, oli tarkoitus korvata osa käytössä olevista FTP-tunnuksista. Samoin Faspexin AD-liitoksen myötä tavoitteena oli saada käyttäjät siirtymään Faspexin käyttöön kaikissa sisäisissä käyttäjältä käyttäjälle tapahtuvassa tiedostonsiirrossa.

Uudesta ympäristöstä oli luotava sellainen kokonaisuus, jonka hallinta on helppoa, uusia työnkuluja on mahdollisuus toteuttaa pienellä vaivalla, eli parametrisoimalla jo toteutettuja työnkuluja uutta käyttötarkoitusta varten. Oleellisena osana oli myös Orchestrator-sovelluksen tarjoamat laajennokset, joiden avulla voidaan luopua muista vanhoista materiaalinkäsittelyyn liittyvistä sovelluksista.

Ympäristön käyttöönoton yhteydessä koko aineistonsiirtoympäristön ympärivuorokautinen valvonta ja vikapäivystys oli tarkoitus siirtää MTV:n ulkopuolelle. Uudistuksen yhteydessä koulutettaisiin päivystyshenkilökunta hallitsemaan ympäristössä tapahtuvat vikatilanteet, sekä dokumentoimaan kaikki ympäristöön liittyvät palvelimet, levyjaot sekä työnkulkuun liittyvät suoritukset.

## 4 SOVELLUSTEN ASENNUS JA RAJAPINNAT

Käyttöjärjestelmäksi ympäristön kaikille palvelimille valittiin RedHat Enterprise Linux 6 [7]. Syynä valintaan oli RedHatilta saatava tuki käyttöjärjestelmätason ongelmien selvityksiin. RedHatin käyttöjärjestelmäpaketit ovat testattuja RedHatin toimesta. Ympäristön ollessa kriittinen, ei haluta riskeerata ympäristön vakautta valitsemalla käyttöjärjestelmäksi esimerkiksi Ubuntu, jonka paketit päivittyvät useammin kuin RedHatin, mutta päivityksiä ei ole välttämättä testattu yhtä kattavasti kuin RedHatin omia paketteja. Samoin MTV:llä on oma paketinhallintajärjestelmä, joka on rakennettu RedHatin tuotteella. Tällä tavoin voidaan määrittää kaikille ympäristön palvelimille samat paketit ja palata helposti päivityksissä tapahtuvissa ongelmissa edelliseen versioon. Kaikki Palvelimet asennettiin VMware-virtuaalialustalle [8].

Sovellusten asennukset suoritettiin yhdessä Asperan asiantuntijoiden kanssa. Sovellusten perusasennukset ovat yksinkertaisia. Jokaisesta asennetusta sovelluksesta on olemassa Windows-, Linux- (Debian, RedHat, Ubuntu), BSD- sekä OSX-asennuspaketit.

### 4.1 Verkkoympäristö

Palvelimet jaettiin kolmeen uuteen verkkoon, jotka on erotettu toisistaan fyysisen palomuurilaitteen avulla. Jaon tarkoituksena on suojata sovelluksia ja palvelemia siten, että julkisesta Internetistä ei ole verkkopääsyä sellaiselle palvelimelle, jolle ei ole tarvetta päästä.

Aspera Connect Server sekä Faspex sijoitettiin ainoina palvelimena DMZ-verkkoon. sillä ne ovat palvelimia, joihin käyttäjät muodostavat yhteyden julkisen Internetin yli.. Yleisesti DMZ-verkkoon sijoitetaan sellaiset palvelimet ja palvelut, joita on tarve käyttää julkisesta Internetistä, kuten DNS, FTP ja web-palvelimet.

Siirtopalvelimille sallittiin palomuurista taulukon 3 mukaiset verkkoyhteydet sisään tulevalle liikenteelle. DMZ-verkosta ei ole rajoituksia ulospäin Internetiin suuntautuvalle liikenteelle. Palomuuariavaukset on ohjeistettu sovellusten asennusohjeissa. Aspera suosittelee käyttämään SSH-protokollan kanssa porttia 33001. Tämä siitä syystä, että perinteistä porttia 22 [9] yritetään skannata verkosta hyökkäyksiä varten. Portin vaihdolla pyritään vähentämään verkkohyökkäyksiä. Samoin Orchestrator ja Asperan muut sovellukset yrittävät oletuksena muodostaa yhteyden 33001 porttiin, kun niiden on tarve suorittaa siirtoja käyttäen Aspera Connect Serveriä.

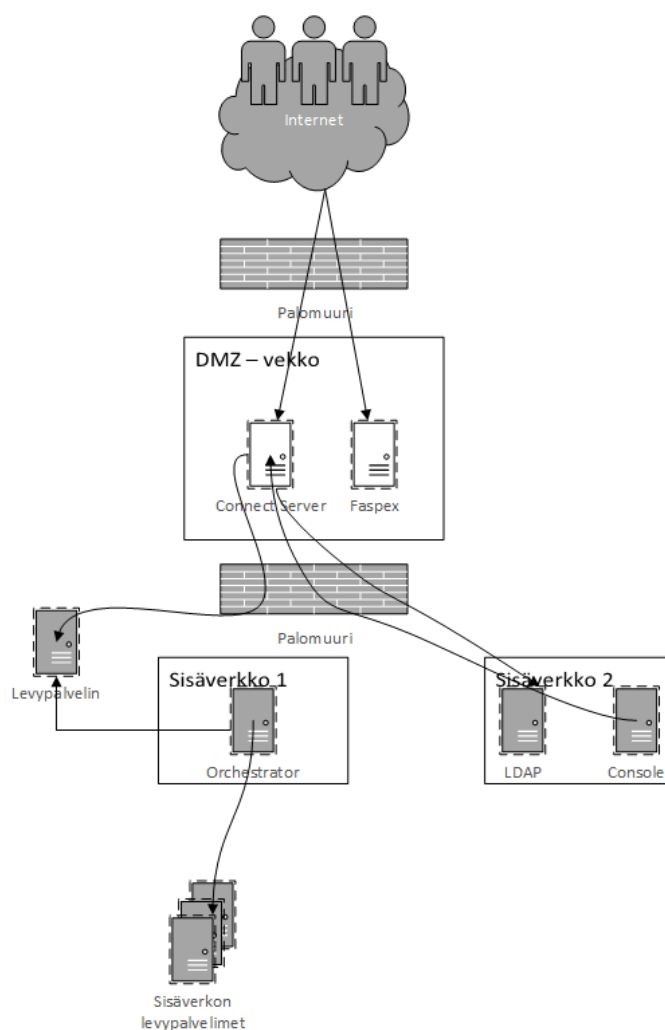
**Taulukko 3: Connect Serverin palomuuuriavaukset**

Portti	Tarkoitus
TCP 33001	SSH, kontrolli- ja autentikointikanava.
UDP 33001 – 33010	Siirtokanava. Sovellus valitsee porttialueelta yhden portin, jota käyttää siirtoon.
TCP 80+443	HTTP-käyttöliittymät

Orchestrator-palvelin sijoitettiin omaan verkkoonsa sen vaatimien levyjakojen takia. Orchestrator-sovellus vaatii työkulkujen takia pääsyä monissa eri verkoissa oleviin levyjakoihin. Jotta saataisiin mahdollisimman tiukasti rajattua pääsy vain ja ainoastaan tarvittaviin levyjakoihin perustettiin Orchestratoria varten oma verkko. Tällä vältettiin se, ettei vanhan skripti-palvelimen tarpeettomia avauksia jäänyt päälle uudessa ympäristössä. Samoin uuden verkon myötä jokaiselle avaukselle löytyi sovellutus, joka kyseistä avausta tarvitsi. Näin saatiin samalla dokumentoitua kaikki avaukset, jotta järjestelmän ylläpito olisi jatkossa helpompaa, mahdollisten verkkomuutosten myötä.

Aspera Console ja LDAP-palvelimet sijoitettiin omaan verkkoonsa. Näille palvelimille ei ole tarvetta päästä kuin järjestelmän ylläpitäjiä. Connect Serverin siirtokäyttäjät luodaan LDAP-palvelimelle. Tätä varten avattiin Connect Serveristä pääsy LDAP-palvelimelle. Samoin Connect Server kirjoittaa siirtojen logia Console-palvelimella olevaan MySQL-tietokantaan. Toinen vaihtoehto olisi ollut sama verkko kuin Orchestrator-sovelluksella. Tavoitteena oli kuitenkin sijoitella palvelimet käyttötarkoituksen mukaan omiin verkkoihin, jonka takia ei nähty ongelmaa luoda kolmatta verkkoa näille palvelimille. Tällä tavalla voitiin myös rajoittaa DMZ-verkosta tapahtuva liikenne eri sisäverkkoihin. Jos LDAP-palvelin olisi ollut samassa verkossa Orchestrator-sovelluksen kanssa, olisi DMZ-verkosta ollut pääsy sekä ympäristön käyttäjänhallintaan, että laajaan määrään eri levyjakoja. Sisäverkon sisällä pääsyjä ei ole avattu, joten tällöin murtamalla yhden palvelimen mahdollinen hyökkääjä ei voisi saada haltuunsa toisia, eri verkoissa olevia palvelimia. Fyysisen palomuurin lisäksi palvelimille asetettiin Linux-käyttöjärjestelmän oma palomuurisovellus iptables [10] päälle. Iptables-palomuurilla saatiin rajoitettua samassa verkossa olevien palvelinten välistä liikennettä.

DMZ-verkon ja sisäverkon välissä käytettävä levypalvelin oli sijoitettu valmiiksi omaan verkkoonsa. DMZ-verkosta ja Orchestrator-sovelluksen verkosta avattiin tarvittavat portit palomuurista, jotta käytetyt CIFS- ja NFS-levyjaot saatiin molemmille palvelimille näkyviin. Uusi verkkoympäristön periaatekuva on esitetty kuvassa 11.



**Kuva 11: Uuden ympäristön verkkoympäristö**

## 4.2 Rajapinnat eri järjestelmiin

Aineistonsiirtoympäristöstä on yhteyksiä eri järjestelmiin. Tärkeimpinä rajapintoina toimivat verkkolevyt, joiden avulla saapuvia tai ulospäin lähteviä tiedostoja siirretään sisäverkon ja DMZ-verkon välillä. Samoin sähköpostilaatikot toimivat muutamassa toteutuksessa materiaalin toimitusväylänä.

Connect Serverin käyttäjähallinta toteutettiin asentamalla erillinen LDAP-palvelin, jonne kaikki Connect Serverin käyttäjät lisättiin. Tämä mahdollisti aikaisempaan ympäristöön verrattuna keskitetymmän paikan hallita käyttäjiä. Aikaisemmin tunnukset olivat paikallisia tunnuksia siirtopalvelimella. LDAP:n avulla pystyttiin jakamaan käyttäjiä omiin ryhmiinsä ja talletettua paremmin tietoa siitä mikä on tunnukset tarkoitus ja kuka on sen hallinnoija.

MTV:n käytössä on oma materiaalin hallintasovellus jota kutsutaan MAM:ksi (Media Asset Management / MAM). Kaikki materiaali indeksoidaan tässä hallintasovelluksessa ja sitä voi ajatella suurena mediapankkina, josta käyttäjät voivat hakea tarvitsemansa materiaalin. Suurin osa Orchestratorin käsittelemistä tiedostoista päättyy lopulta MAM:iin. MAM:ssa on omat työnkulut, jotka käsittelevät saapuneet tiedostot useaan eri muotoon. Orchestrator toimittaa MAM:iin päätyvän materiaalin MAM:in tarkkailemiin hakemistoihin, josta ne siirtyvät MAM:in toimesta eteenpäin toimittajien käyttöön.

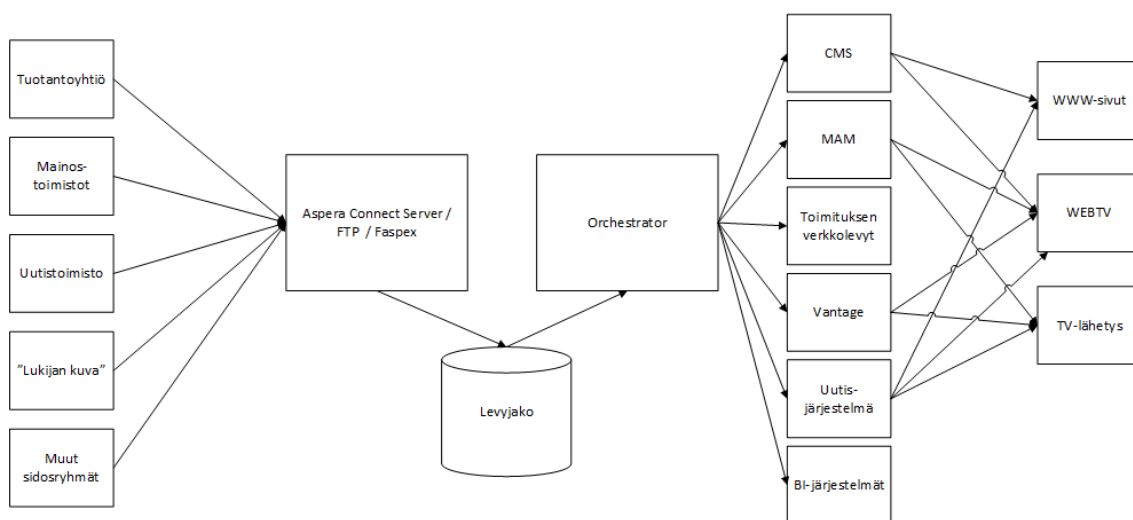
Joissain työkuluissa suoritetaan REST-kutsuja ulkoisiin järjestelmiin sen jälkeen kun tiedostot on ensin siirretty kohdejärjestelmään. Nämä tapaukset ovat yleisesti sellaisia, että MTV toimittaa omaa materiaaliaan toiselle toimijalle. REST-kutsun avulla ilmoitetaan kohdejärjestelmään, että uutta materiaalia on toimitettu. Joissain työkuluissa Orchestrator hakee REST-kutsulla tietystä URL:sta dataa ja prosessoi sen eteenpäin toisen järjestelmän käyttöön käyttäen XSLT-muunnosta tai käsin kirjoitettua Ruby-koodia, esimerkiksi muodostetaan sivulla olevasta HTML-taulukosta CSV-tiedosto.

Myös MTV:n käyttämä CMS (Content Management System), eli web-sivujen julkaisu- ja sisällönhallintajärjestelmä toimi yhtenä rajapintana. Esimerkiksi ulkopuolisella käyttäjällä on mahdollisuus ladata oma uutiskuva erillisen palvelun kautta. Orchestrator suorittaa tässä välissä tiedostolle tarkistuksen ja siirtää kuvan edelleen CMS-järjestelmän käyttöön. Toimittaja pystyy näin käyttämään ”Lukijoiden kuvia” osana uutisartikkelia.

Transkooderit toimivat ohjelmamateriaalin toimittamisen työnkulussa tärkeänä rajapintana ympäristön ja käyttäjän välillä. Transkooderit muuntavat sisääntulleeseen videomateriaalin TV-lähetykseen sopivaksi. Saapuva ohjelmamateriaali on yleisesti sellaisessa muodossa, ettei sitä voida suoraan käyttää lähetyksjärjestelmässä. Esimerkiksi audiokanavia tarvitsee miksata eri kanaville tai ääniraitojen äänentasot pitää tasata. Tällöin sisääntuleva materiaali muunnetaan eli, transkoodataan TV-lähetykseen sopivaksi.

MTV:n käytössä on Vantage transkoodausjärjestelmä [11]. Vantage tarjoaa rajapinnan, jonka avulla sitä voidaan kutsua ulkoisista järjestelmistä. Orchestratorissa on valmiina Vantagen rajapinnan käyttöön suunniteltu lisäosa. Vantagen toiminta perustuu myös työnkulkuihin. Työnkululle annetaan parametrina transkoodattava videotiedosto sekä parametrit, joiden mukaan työnkulku säättää audiokanavien tasot sopiviksi. Vantagen käyttö suoraan Orchestratorin kautta oli yksi projektin suurimmista toteutuksista. Tämä toteutus on avattu luvussa 5.

Eri rajapintoja havainnollistaa kuva 12, jossa on pyritty kuvaamaan eri järjestelmät, joihin aineistonsiirtoympäristö on yhteyksissä. Vasemmalla puolella kuvaa on listattuna eri sidosryhmiä, jotka toimittavat materiaalia. Vastaavasti oikealla puolella on ensin sisäisiä järjestelmiä, joihin aineistonsiirtoympäristö toimittaa materiaalin, joko valmiiksi julkaisukelpoisessa muodossa tai välittää sen eteenpäin jatkokäsittelyyn. Näiden jälkeen tulevat julkiset, loppukäyttäjille näkyvät palvelut.



**Kuva 12: Aineistonsiirtoympäristön integraatiot**

### 4.3 Aspera Connect Server sovelluksen asennus

Aspera Connect Server asennetaan RedHat-käyttöjärjestelmälle suoraan *rpm*-paketista [12]. Connect Serverillä ei ole riippuvuuksia muista sovelluksista.

Connect Serverin käyttäjänhallinta muodostuu kahdesta osasta:

- palvelimella olevista normaaleista käyttöjärjestelmän käyttäjätunnuksista
- Connect Serverin omasta sisäisestä käyttäjähallinnasta.

Connect Server yhdistää käyttöjärjestelmän käyttäjätunnuksen osaksi omaa käyttäjähallintaansa. Käyttäjän salasana sovelluksen käyttöön on sama kuin käyttöjärjestelmän salasana. Connect Serverin asetuksissa määrätään käyttäjälle levyllä oleva hakemisto, joka toimii käyttäjän kotihakemistona. Käyttäjälle ei ole oikeutta päästä tämän hakemiston ulkopuolelle, mutta käyttäjä voi luoda uusia hakemistorakenteita kyseisen kotihakemiston alle.

Samoin käyttäjälle voidaan määrittää muita haluttuja asetuksia, kuten rajoittaa käyttäjän maksimaalista siirtonopeutta. Kaikki mahdolliset asetukset on kuvattu Connect Serverin asennusohjeessa [13].

Käyttöjärjestelmätasolla käyttäjän shell-komentotulkiksi asetetaan Asperan oma *asps*shell joka ei salli komentotulkin ajoa. Tällöin käyttäjällä ei ole oikeutta kirjautua palvelimelle, vaan Asperan sovellus ainoastaan autentikoi käyttäjän käyttäen Linuxin autentikointimekanismeja.

Käyttäjän autentikointi toimii kokonaisuudessaan seuraavasti:

- 1) ennen varsinaista siirtoa käyttäjän asiakasohjelma muodostaa SSH-yhteyden palvelimeen
- 2) SSH-yhteys autentikoidaan joko käyttäen salasanaa tai SSH-avainparia
- 3) palvelinsovellus varmistaa, että käyttäjätunnusta vastaava tunnus löytyy myös Connect Serverin omasta käyttäjähallinnasta



- 4) yhteys muodostuu ja käyttäjän on mahdollista ladata tiedostoja palvelimelle tai palvelimelta.

Kaikki Connect Serverin ja FTP:n käyttäjätunnukset luotiin asennetulle LDAP-palvelimelle. Projektin alkuvaiheessa tarkoituksena oli yhdistää Faspex-sovelluksen käyttäjähallinta samaan LDAP-järjestelmään. Tällöin samat tunnukset olisivat toimineet sekä Connect Serverissä että Faspexissa. Toteutusta muutettiin kuitenkin siten, että Faspex liitettiin osaksi MTV:n omaa Active Directory-järjestelmää. Tällöin käyttäjät voivat käyttää Faspexia samoilla tunnuksilla kuin MTV:n omia järjestelmiä. Faspex-tunnukset liittyivät pääsääntöisesti sisäisiin tiedoston siirtoihin ja Connect Serverin käyttö ulkoisten toimijoiden toimituksiin.

#### 4.4 Aspera Orchestrator

Orchestrator asennetaan Connect Serveristä poiketen osittain käsityönä. Orchestratorin asennuspaketti siirretään tiettyyn hakemistoon palvelimella, jonka jälkeen suoritetaan Bash-skripti joka luo tarvittavat käyttäjät ympäristöön, sekä suorittaa tietokannan konfiguroinnin sovelluksen käyttöön. Orchestrator tallettaa työnkulut sekä niiden suoritusten tilastot tietokantaan. Tämän takia sovellusta varten asennettiin MySQL-tietokanta [14]. MySQL-tietokannan voi asentaa joko Asperan omasta paketista tai käyttöjärjestelmän oman paketinhallinnan kautta. Orchestrator tarjoaa myös mahdollisuuden käyttää myös Oracle- tai MSSQL-tietokantoja. Projektin alkaessa Oracle- tai MSSQL-tietokantojen käyttömahdollisuutta ei vielä ollut, joten asennus tehtiin käyttämällä MySQL-tietokantaa.

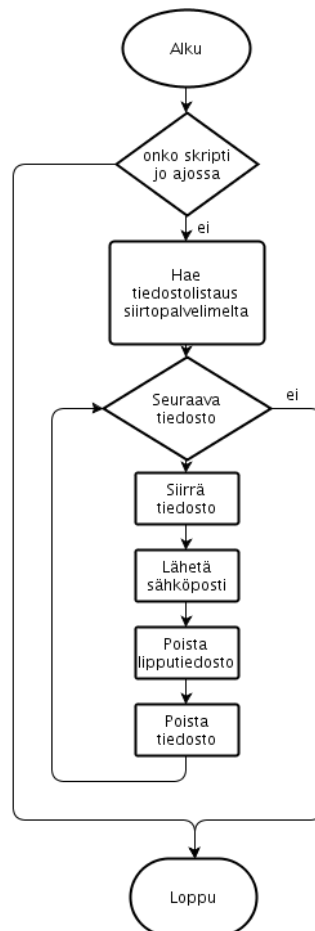
Sovellusasennuksen lisäksi suurin työ asennuksen kanssa oli tarvittavien levyjakojen tuonti palvelimelle. Lähes kaikki alkuvaiheessa toteutetut työnkulut keskittyivät tiedostonsiirtoon DMZ-verkosta sisäverkkoon. Tätä varten käytiin läpi vanhan skripti-palvelimen kaikki levyjaot ja laadittiin uudet palomuurisäännöt siten, että vastaavien levyjakojen tuonti uuteen Orchestratorin käytössä olevaan verkkoon oli mahdollista. Samalla pyrittiin hieman yhtenäistämään jakoja. Tarkoituksena oli välttää samasta levyjaosta useamman hakemiston tuontia erikseen, vaan tuoda jaosta juurihakemisto ja näyttää tämä jako ainoastaan Orchestratorin palvelimelle.

## 5 TYÖNKULKUJEN RAKENTAMINEN

Työnkulkuja materiaalin siirtämiseen lähdettiin rakentamaan käymällä läpi olemassa olevat Bash-skriptit. MTV:n kanssa käytiin yhdessä läpi skriptien käyttötarkoitus sekä tarpeellisuus. Tarpeellisia työnkulkuja lähdettiin toteuttamaan Orchestrator-sovellukseen.

### 5.1 Bash-skriptien siirto työnkuluiksi

Lähes kaikki alkuperäiset Bash-skriptit olivat periaatteeltaan kuvan 13 mukaisia.



*Kuva 13: Bash-skriptien periaate*

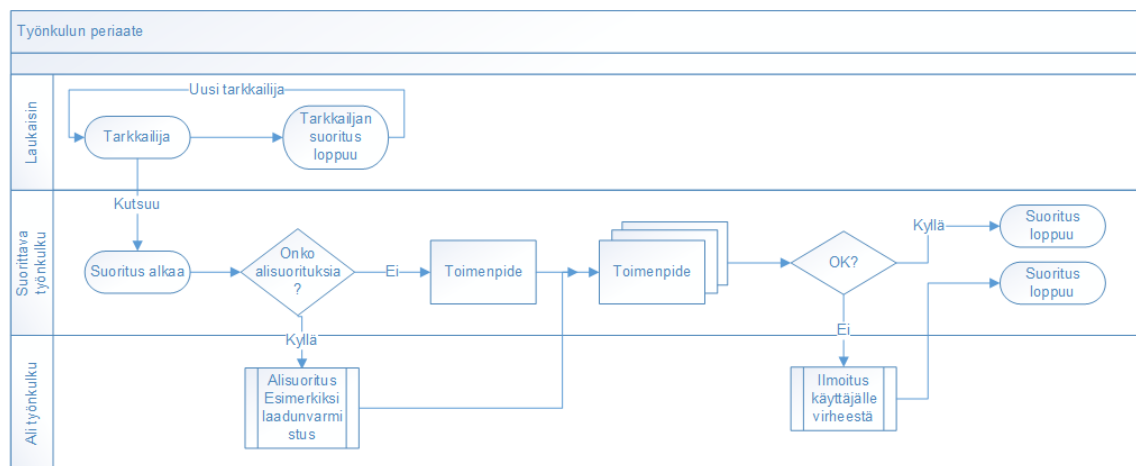
- tarkastus, onko suoritus jo käynnissä
- tunnistetaan saapunut tiedosto
  - tunnistaminen tehtiin vertailemalla edellisen ajon tiedostolistausta uuden suorituksen tiedostolistaukseen.
- muodostetaan tiedostosta lipputiedosto, jotta sitä ei käsitellä kahteen kertaan
- tiedoston siirto DMZ-verkosta sisäverkkoon
  - tiedoston kopiointi tehtiin rsync-sovelluksella
- alkuperäisen tiedoston poistaminen DMZ-verkkoon näkyvältä levyiltä
- lipputiedoston poisto
- sähköposti-ilmoitus.

Työnkuluissa saattoi olla muitakin toiminnallisuuksia, mutta peruseriaate oli kaikissa sama. Työnkuluista luotiin kuvan 13 mukaiset lohkokaaviot, jotta niiden toiminnallisuus saatiin avattua. Lohkokaaviot muistuttavat lähes suoraan Orchestrator-työnkulun toteutusta, jolloin skriptin kopiointi Orchestrator-sovellukseen työnkuluksi helpottui.

MTV:n vastuuhenkilö toimi jokaisen uuden työnkulun määrittelijänä ja loi jokaisesta tarvittavasta työnkulusta vaatimusmäärittelyn. Vaatimusmäärittelyssä kuvattiin työnkulkuun liittyvät levyjaot (lähde- ja kohdehakemistot), yhteyshenkilöt, lokitukseen liittyvät vaatimukset, sekä lyhyt kuvaus mihin kyseistä työnkulkua käytetään.

Yksittäistä työnkulkua lähdettiin rakentamaan luomalla ensin uuteen ympäristöön hakemistot saapuville tiedostoille. Vastaavasti luotiin kohdehakemistoon testihakemisto, jotta ei sekoitettaisi tuotantojärjestelmää testiajoilla.

Työnkulut rakennettiin pääosin kuvan 14 mukaisesti.



**Kuva 14: Työnkulun periaate**

Työnkuluissa laukaiseva (tarkkailija) työnkulku kutsuu suorittavaa työnkulkua, jonka jälkeen muodostuu uusi, itsenäinen työnkulku, joka suorittaa yhteen tiedostoon liittyvät tehtävät. Tämä sen takia, että voidaan vapauttaa laukaiseva työnkulku jatkamaan uusien tiedostojen saapumista, eikä sen tarvitse odottaa yhden tiedoston käsittelyn valmistumista.

Suorittava työnkulku suorittaa määritellyt tehtävät:

- tarvittavien alityönkulkujen kutsuminen
  - esimerkiksi laadunvarmistus
- tiedoston siirto kohdehakemistoon
- sähköposti-ilmoitus
- lokitiedoston kirjoitus.

Jos työnkuluissa ei tapahtunut virheitä, työnkulku päättyy onnistuneesti

Jos alityönkulussa tapahtuu virhetilanne, esimerkiksi laatu tiedot eivät täytä kriteerejä, palautetaan alityönkulusta suorittavaan työnkulkuun virheilmoitus ja käsitellään tällaiset tiedostot erikseen omassa haarassaan.

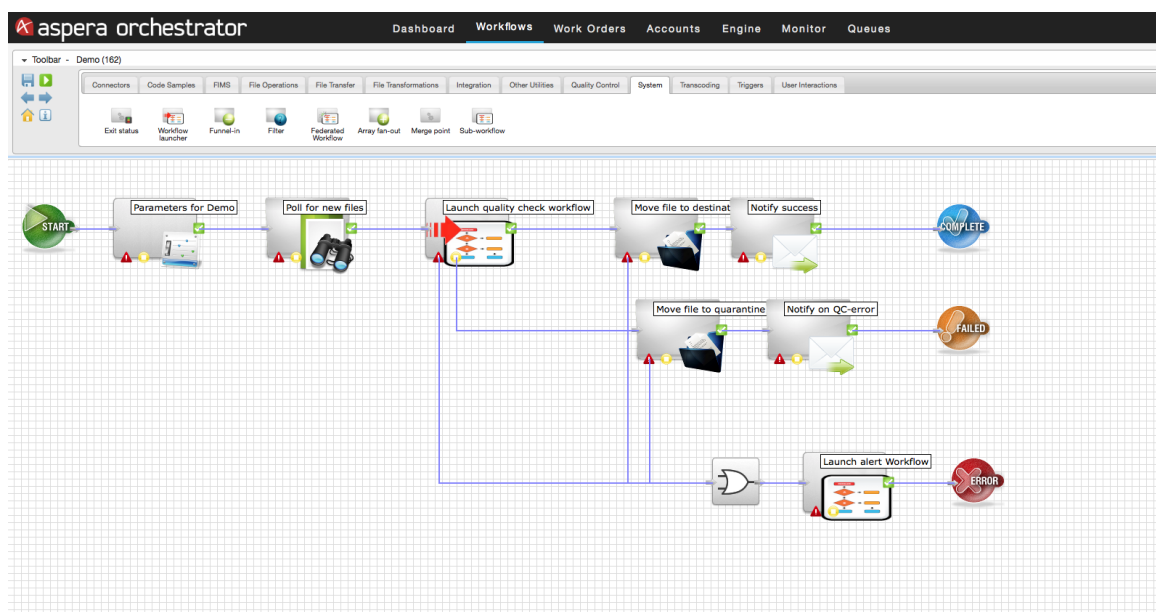
Työnkulkuja lähdettiin rakentamaan siten, että jokaista siirtoa kohden on oma laukaiseva työnkulku. Mahdollisuus olisi ollut myös luoda vain muutama laukaiseva työnkulku ja tämän jälkeen päätellä saapuneen tiedoston hakemistosta, mihin toteutukseen kyseinen tiedosto kuuluu ja laukaista sen jälkeen varsinaisen tiedoston käsittelyn hoitava työnkulku. Valinta tehtiin erillisten laukaisevien työnkulkujen eduksi sen takia, että näin saatiin luotua selkeä näkyvyys sille, onko tiettyyn tunnukseen tai kokonaisuuteen liittyvä työnkulku käynnissä. Samoin tällä vältettiin se, että yhden tarkkailijan virhe ei aiheuta ongelmia muihin työnkulkuihin.

Orchestratorin mukana tuomia uusia ominaisuuksia käytettiin aina hyväksi, kun se oli mahdollista tai tarkoituksenmukaista. Monissa tapauksissa tiedostolle tehtiin jatkokäsittelyä erillisen järjestelmän, kuten transkooderin toimesta. Transkooderin asetukset on yleisesti määritelty siten, että sille voidaan tuoda vain tietyn tyyppisiä tiedostoja. Vanhassa järjestelmässä oli mahdollista, että pelkästään videoaineiston toimittamiseen tarkoitetun kanavan kautta toimitettiin myös kuvamateriaalia. Vanhan järjestelmän siirtäessä kuvatiedoston transkooderin käsiteltäväksi transkoodaus luonnollisesti epäonnistui. Tällaiset virheelliset tiedostot kuluttavat turhaan resursseja.

Uuden työnkulun myötä saapuville tiedostoille tehtiin yksinkertainen tarkistus, että saapunut tiedosto todellakin on sellaista muotoa, jota vastaanottava järjestelmä pystyy käsittelemään ja vasta tämän jälkeen siirrettiin tiedosto seuraavaan järjestelmään. Virheelliset tiedostot siirretään työnkulun toimesta erilliseen karanteeni-hakemistoon, josta käyttäjä pystyy ne tarvittaessa hakemaan ja todentamaan, että tiedosto on todellakin virheellinen.

## 5.2 Työnkulun rakentaminen Orchestrator-sovelluksessa

Työnkulkuja rakennetaan Orchestrator-sovelluksessa liittämällä eri toiminnallisuuksia toisiinsa kuvan 15 mukaisella tavalla.



**Kuva 15: Esimerkki Orchestratorin työnkulusta**

Työnkulun käynnistää laukaisin. Erityyppiset laukaisimet on avattu luvussa 2.3. Jokainen lohko tarjoaa yhden sisääntulon sekä kolme ulostuloa. Riippuen lohkon toiminnallisuudesta virhetilanne palauttaa joko ”Failed” tai ”Error” virheilmoituksen kanssa. Jos lohkon suoritus onnistuu, palauttaa se ”Complete”. Suorituksen lisäksi lohko palauttaa ulostulossaan joukon arvoja. Arvot riippuvat lohkon toiminnasta. Esimerkiksi uusien tiedostojen tarkkailija palauttaa ulostulossaan ”Complete” sekä löytyneet tiedostot täydellisine polkuineen, kun uusia tiedostoja on ilmestynyt sen tarkkailemaan hakemistoon. Vastaavasti jos tarkkailtavaa hakemistoa ei ole olemassa palauttaa tarkkailija tilan ”Failed” sekä virheilmoituksen, jossa kerrotaan että määritettyä hakemistoa ei löytynyt. Error- ja failed ulostuloihin määritellään käsittely miten virhetilanteissa toimitaan.

Modulaarisen rakenteen ansiosta Orchestrator-sovellukseen on mahdollista luoda alityönkulkuja, joita voidaan käyttää useissa eri käyttötapauksissa. Alityönkulku palauttaa normaalin lohkon tapaan tiedon siitä, oliko suoritus onnistunut vai tapahtuiko suorituksessa virheitä. Alityönkulun perusteella voidaan jatkaa varsinaisen työnkulun suoritusta halutulla tavalla.

Yleisin käytetty alityönkulku on saapuneen tiedoston laadunvarmistus. Riippuen toteuttavasta toiminnallisuudesta laadunvarmistuksen toteuttavaan työnkulkuun määritetään kutsuvasta työnkulusta tarvittavat laadunvarmistuksen parametrit, esimerkiksi videotiedoston pienin sallittu kuvataajuus tai kuvan resoluutio. Vastaavanlaisen laadunvarmistuksen toteuttaminen ei olisi ollut mahdotonta vanhassa ympäristössä, mutta se olisi kuitenkin vaatinut paljon enemmän työtä verrattuna siihen, että Orchestrator-

sovelluksesta löytyy valmiit lohkot laadunvarmistuksen työkaluihin, kuten Mediainfo-sovelluksen palauttamien arvojen vertailemiseen ennalta määrättyihin raja-arvoihin.

## 6 OHJELMAMATERIAALIN KÄSITTELYN UUSIMINEN

Sisään tulevan ohjelmamateriaalin käsittely oli vanhoista toteutuksista liikennemäärältään suurin, samoin ohjelmamateriaalin käsittelyn hoitavalla työkululla oli kaikista eniten eri käyttäjiä. Käyttäjät olivat pääsääntöisesti tuotantoyhtiöitä, jotka siirsivät Asperan FASP:n tai FTP:n kautta ohjelmamateriaalia MTV:n käyttöön. MTV:n materiaalinkäsittelijät saattoivat myös hakea tuotantoyhtiön palvelimelta ohjelmamateriaalia ja siirtää sen käsittelijän käsiteltäväksi manuaalisesti. Ohjelmiin liittyvää materiaalia tulee päivittäin 20–30 tiedostoa, tiedostojen koon ollessa keskimäärin noin 40 gigatavua.

Ohjelmamateriaalin käsittelyssä oli eri vaiheita ja jokaiseen eri vaiheeseen oli oma sovelluksensa. Tarkoitus oli yhtenäistää käsittely siten, että eri sovellukset korvattaisiin täysin Orchestratorilla.

### 6.1 Alkuperäinen toteutus

Alkuperäinen toteutus oli toteutettu käyttäen Bash-skriptejä, erillisistä QCTools-sovellusta sekä Vantage-transkoodereita. QCTools oli MTV:n käyttöön luotu sovellus, joka tarjosi mahdollisuuden saapuneen materiaalin käsittelyyn sekä kirjanpitoon.

Materiaalinkäsittelyn alkuperäinen toteutus oli seuraava:

1. Bash-skripti tunnisti tunnuskohtaiseen DMZ-verkossa sijaitsevaan hakemistoon saapuneen tiedoston.
2. Skripti siirtää tiedoston kohdelevylle sisäverkkoon tunnuskohtaiseen QCToolsin tarkkailemaan hakemistoon.
3. QCTools tunnistaa saapuneen tiedoston, lisäten tiedoston nimeen aikaleiman.
4. Materiaalinkäsittelijä valitsee QCToolsin käyttöliittymän kautta tiedoston, jonka haluaa käsittelemään.
5. QCTools luo tiedostosta linkin Vantage-transkoodausjärjestelmän tarkkailemaan hakemistoon.
6. Vantage suorittaa transkoodauksen saapuneelle tiedostolle, jotta siitä saadaan TV-lähetykseen sopiva versio.
7. Käyttäjä saa ilmoituksen transkoodauksen valmistumisesta.
8. Käyttäjä tarkistaa mediasoittimella transkoodauksen laadun sekä audiokanavien oikeellisuuden.

Alkuperäisessä toteutuksessa oli käyttäjien sekä ylläpidon kannalta muutamia puutteita ja ongelmia. Käyttäjän ei ollut mahdollista vaikuttaa siihen, missä järjestyksessä tiedos-

tot saapuivat QCTools-sovellukseen. Eteen saattoi tulla tilanne, jossa paljon materiaalia saapui lähes yhtäaikaaisesti Bash-skriptin tarkkailemaan hakemistoon. Skripti siirsi tiedostoja yksi kerrallaan vanhimmasta alkaen. Jos samanaikaisesti tuli materiaalia, joka piti saada jatkokäsittelyyn mahdollisimman nopeasti, ei materiaalinkäsittelijällä ollut mahdollisuutta vaikuttaa tähän. Ainoa tapa saada kiireellinen materiaali jatkokäsittelyyn nopeammin, oli ottaa yhteys sovellustukeen, joka siirsi kiireellisen materiaalin käsin.

Toinen ongelma oli, ettei materiaalinkäsittelijöillä ollut suoraa pääsyä lokitiedostoihin, jolloin he eivät olleet varmoja siitä oliko haluttu tiedosto saapunut jo palvelimelle odotamaan käsittelyä. Skripti lähetti sähköposti-ilmoituksen vasta sen jälkeen kun tiedosto oli kopioitu pois DMZ-verkosta sisäverkon palvelimelle.

Käyttäjillä ei myöskään ollut näkyvyyttä Vantage-transkoodauksen etenemiseen QCTools-sovelluksen kautta. Eli käyttäjä ei voinut mitenkään arvioida kuinka kauan videon transkoodauksessa tulee kestämään. Vantage-järjestelmää käyttävät lähes kaikki MTV:n eri yksiköt. Tämän takia Vantageen saattoi kerääntyä jonoa, joka purkaantui sitä mukaa kun Vantage sai tehtäviä suoritettua. Materiaalinkäsittelijän oli otettava yhteyttä Vantagen ylläpitoon, jos hän halusi oman tiedostonsa jonon kärkeen.

Järjestelmäylläpidon puolelta katsottuna tiedostojen kopioiminen DMZ-verkosta sisäverkkoon muodosti turhan välivaiheen. Vantagen transkoodaama lopullinen videotiedosto talletettiin ainoastaan sisäverkkoon näkyvälle palvelimelle. Materiaalia liikkui järjestelmän kautta parhaimmillaan useita satoja gigatavuja päivässä, jonka takia tiedostojen kopiointi DMZ-verkosta sisäverkkoon ennen transkoodausta nähtiin turhana välivaiheena, joka kuormitti sisäverkkoa turhaan.

## 6.2 Uuden toteutuksen ensimmäinen vaihe

Jotta vanhasta skriptipalvelimesta päästäisiin mahdollisimman nopeasti eroon, oli tarkoitus ensin siirtää ainoastaan Bash-skriptin suorittama osuus Orchestrator-sovellukseen ja vasta tämän jälkeen miettiä, kuinka saataisiin käytettyä hyväksi Vantagen rajapintoja ja siirtämällä QCToolsin toiminnallisuudet myös Orchestratoriin.

Bash-skriptin siirtämisessä Orchestratoriin ei ollut ongelmia, vaan toteutus oli erittäin suoraviivainen. Toteutukseen lisättiin kuitenkin hieman toiminnallisuutta, jotta loppukäyttäjille tulisi mahdollisuus tutustua Orchestratorin käyttöliittymään, sillä jatkossa materiaalinkäsittely tulisi tapahtumaan täysin Orchestratorin kautta.

Saapuneiden tiedostojen käsittely toteutettiin seuraavasti:

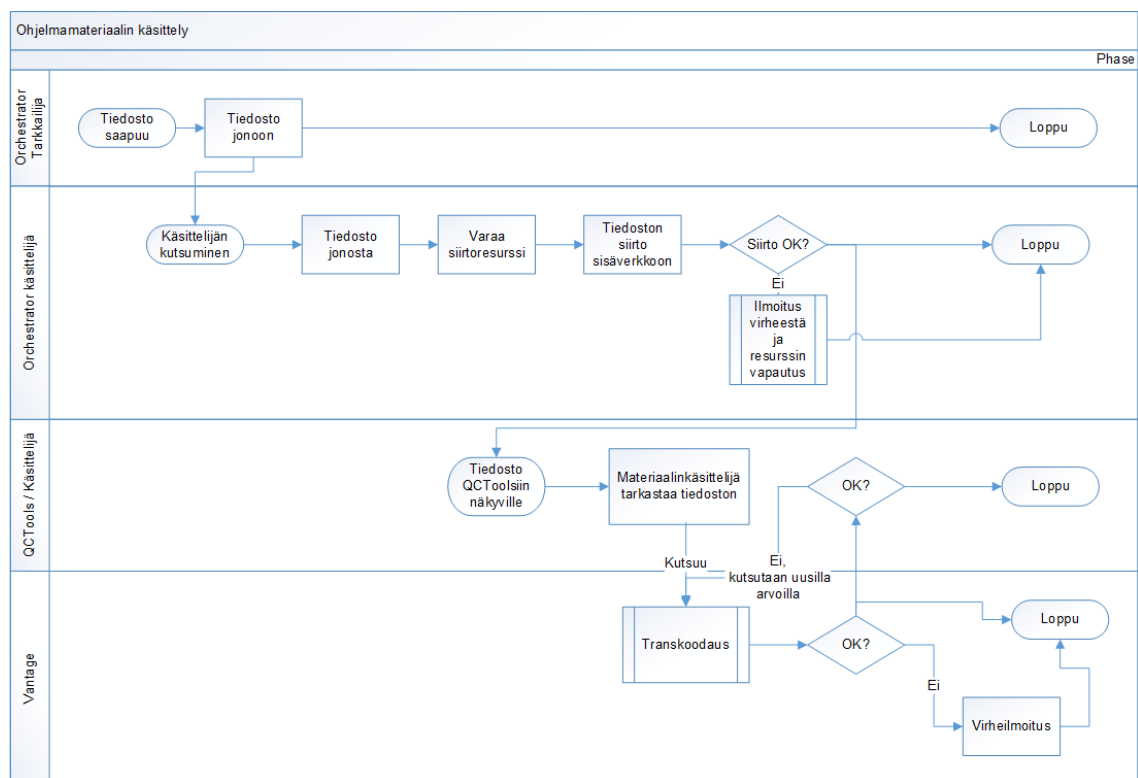
- tarkkailija tunnisti saapuneet tiedostot, ja tästä kirjoitettiin logitiedostoon merkintä
- saapuneet tiedostot sijoitettiin jonoon, joka toimi periaatteella: ensimmäinen sisään tullut tiedosto käsiteltäisiin ensimmäisenä
- aina kun tiedosto meni jonoon, kirjoitettiin Orchestratorin lokiin merkintä saapuneesta tiedostosta



- tämä mahdollisti tiedostojen etsimisen virhetilanteissa
- käyttäjille annettiin Orchestratorin kautta mahdollisuus muokata jonossa olevia tiedostoja, siirtelemällä niitä jonossa ylös- tai alaspäin.
  - tämä mahdollisti tiedostojen priorisoinnin, mitä aikaisemmassa toteutuksessa ei ollut
- jonosta siirrettiin tiedosto QCTools-sovelluksen tarkkailemaan hakemistoon
- QCToolsista materiaalinkäsittelijä valitsi tiedostot, jotka lähtivät eteenpäin transkoodattavaksi Vantageen.

Vantagen työnkulku oli rakennettu kahteen eri osaan: Laadunvarmistus- ja audioanalyysi ja varsinainen transkoodaus. Tarkastus- ja audioanalyysityönkulussa Vantage suoritti tiedostolle laadunvarmistuksen, sekä tarkasteli minkä laatuista materiaalia tiedosto on. Esimerkiksi kuvan resoluution perusteella pääteltiin onko video HD-laatuista. Tarkistuksen jälkeen suoritettiin varsinainen transkoodaus, jonka lopputuloksena tiedosto ilmestyi videopalvelimelle. Täällä materiaalinkäsittelijä teki vielä tarkistuksen tiedostolle ja varmisti videotiedoston eheyden. Jos tiedostossa oli ongelmia, materiaalinkäsittelijä pystyi käynnistämään transkoodauksen uudelleen QCToolsin kautta.

Periaate on esitetty kuvassa 16.



**Kuva 16: Ohjelmamateriaalin käsittelyyn käytetyn työnkulun rakenne.**

Käyttäjille luotiin Orchestratoriin kuvan 17 mukainen työpöytä, josta he näkivät heti oleelliset asiat:

- viimeksi saapuneet tiedostot
- jonon tilanteen, sekä jonon kontrollipainikkeet

- siirtojen tilanne DMZ-verkosta sisäverkkoon
- käsiteltyjen tiedostojen listaus.

The screenshot displays a software interface with two main panels. The left panel, titled 'Assigned tasks for MTV3-QC', lists several tasks: 'Vantage transcoded file: V\_M0403045.mxf', 'V\_M0403044.mxf', 'V\_M0403043.mxf', 'V\_M0403047.mxf', 'V\_M0403046.mxf', 'V\_M0400594\_U.mov', and 'V\_M0399248.mxf'. Below this is a 'File Journal View' section with a search bar and a list of files including '/mnt/program/incoming/mtvohjelmat/M0399248\_U', '/mnt/program/incoming/mtvohjelmat/MythBuster:', '/mnt/program/incoming/hdpost/ V\_M0400594\_uusi', and another 'MythBuster:' file.

The right panel, titled 'Step Vantage transcoding Activity', contains a table with three columns: 'Vantage Workflow', 'Transcoded file', and 'Status'. It lists six transcoding jobs, all using the 'PGM\_IN\_HD\_Priority\_Normal' workflow. Each job shows the source file path (e.g., '\\hans.mtvmedia.netti\program\vantage\V\_M0398985.mxf') and its status, including 'Job started at' and 'Progress made as of' timestamps, with a consistent 61% progress level.

At the bottom of the right panel, it states 'Last refreshed at: 2015-04-28 14:41:25'.

**Kuva 17: Käyttäjän työpöytä**

Tämä kattoi siis luvussa 6.1 olleen alkuperäisen toteutuksen kohdat 1 ja 2.

### 6.3 Lopullinen toteutus

Lopullista toteutusta lähdettiin rakentamaan yhdessä materiaalinkäsittelijöiden kanssa. Tavoitteena oli korvata QCToolsin toiminnallisuudet Orchestratorin ominaisuuksilla ja tätä kautta päästä eroon ylimääräisestä työkalusta materiaalinkäsittelyssä. Päämääränä oli rakentaa käyttäjälle selkeä ja helposti käytettävä käyttöliittymä Orchestratoriin, jonka kautta materiaalinkäsittelijä pystyi tarkkailemaan ohjelmamateriaalin kulkua koko putken läpi.

Työ aloitettiin määrittelemällä työnkulku periaatteellisella tasolla lohkokaaavioksi, jossa oli kuvattuna työnkulun eri vaiheet ja tietyn vaiheen suorittavat ohjelmat tai käyttäjät.

### 6.3.1 Tiedoston saapuminen ja laadunvarmistus

Työnkulku käynnistyy, kun tiedosto saapuu Orchestratorin työnkulun tarkkailemaan hakemistoon. Tiedostolle suoritetaan Mediainfo-sovelluksella laadunvarmistus. Mediainfo palauttaa tiedoston laatutiedot kuten resoluution, formaatin sekä muita videotiedostoon liittyviä ominaisuuksia. Jos mediainfo-sovellus ei palauta tiedostolle laatutietoja, on se merkki siitä, että tiedosto on korruptoitunut siirrossa ja sitä ei voida käyttää. Tästä ilmoitetaan käyttäjälle, joka varmistaa tilanteen ja lataa tiedoston uudelleen palvelimelle.

Mediainfon tietojen perusteella luodaan käyttäjälle tehtävä Orchestratoriin, jossa näytetään tiedoston laatutiedot. Tehtävien tarkoituksena on toimia QCToolsin korvaajana. Orchestratorin tehtäviä voi rakentaa käsin ja niihin voi määritellä tarvittavia sisään- ja ulostuloja. Tehtäviin voi liittää myös käyttäjän valintoja, joiden perusteella voidaan haarauttaa työnkulkua käyttäjän valitsemalla tavalla.

Käyttäjälle osoitetut tehtävät näkyvät erillisessä listassa, josta käyttäjän on mahdollista valita haluttu tiedosto käsittelyyn. Tämä lista toimii samalla jonona. Saapuneet tiedostot odottavat tässä jonossa niin kauan kunnes käyttäjä valitsee sen käsittelyyn. Tämä mahdollistaa suuremmalla prioriteetilla tulevien tiedostojen aikaisemman käsittelyn.

Käyttäjän valitessa tiedoston jonka haluaa käsittelyyn, avautuu käyttäjälle kuvan 18 mukainen käyttöliittymä.

Käyttäjältä kysytään seuraavat tiedot:

- haluttu toiminto
  - siirrä tiedosto transkoodaukseen
  - poista saapunut tiedosto
- ohjelmamateriaalin sisäinen tunniste
- audiokanavien järjestys
- kuvajärjestys (Field Order)
- transkoodauksen prioriteetti.

Pakollisena syötteenä on ohjelmamateriaalin sisäinen tunniste. Tätä kenttää käytetään myös kun käyttäjä haluaa poistaa tiedoston. Tällöin käyttäjä syöttää kenttään syyn, jonka takia hän haluaa tiedoston tuhota.

**Select Vantage profile**

File name: **M0396132.mov**  
 Format: **MPEG-4**  
 Audio channels: **2channels**  
 Duration: **00:40:01.040**  
 Size: **34.14 GiB**  
 Folder: **mtvohjelmat**  
 Mediainfo: **M0396132.mov**  
 Video: **/program/processing/M0396132.mov**

**Please notice!**  
 Default profile assumes, that left channel is on audio track 1 and right channel on audio track 2. Normal field order is used

When selecting custom profile, please provide either Stereo, Dolby, or SMPTE audio profile.  
 Also please provide values for left and right channel when selecting "Stereo" option  
**When selecting delete, please provide your name on Internal name field for logging purposes**

Select action: Transcode with following values

**Provide Internal ID for this asset (example: M1735644)**

**Select audio options for encoding, not needed when Default profile is selected**

☒ Stereo  
 1 Left Channel  
 2 Right Channel  
☐ 5.1 Dolby  
☐ 5.1 SMPTE

**Select field order**  
☒ Normal field order  
☐ Reversed field order

Select priority: Normal

Submit Cancel

### Kuva 18: Käyttäjälle näytettävä tehtävä

Vastaavat asiat oli toteutettuna QCToolsiin lukuun ottamatta transkoodauksen prioriteetin valintaa. Toteutuksen yhteydessä valinnat siirrettiin osaksi Orchestratorin työnkulkua, jolloin käyttäjän ei tarvitse erikseen avata QCToolsia hallitakseen materiaalivirtaa.

Mediainfon käyttäminen laadunvarmistukseen vähentää myös Vantage-järjestelmän kuormaa. Aikaisemmin Vantage suoritti videotiedoston laadunvarmistuksen. Mediainfo-sovelluksella ei voi tehdä laajamittaista laadunvarmistusta. Jos tiedosto on rikkoutunut siirrossa, on sen metatiedot lukukelvottomia. Näihin tilanteisiin tehtiin erillinen käsittely jossa käyttäjälle ilmoitetaan virheestä ja käyttäjä pystyy käsin siirtämään tiedoston jatkokäsittelyyn.

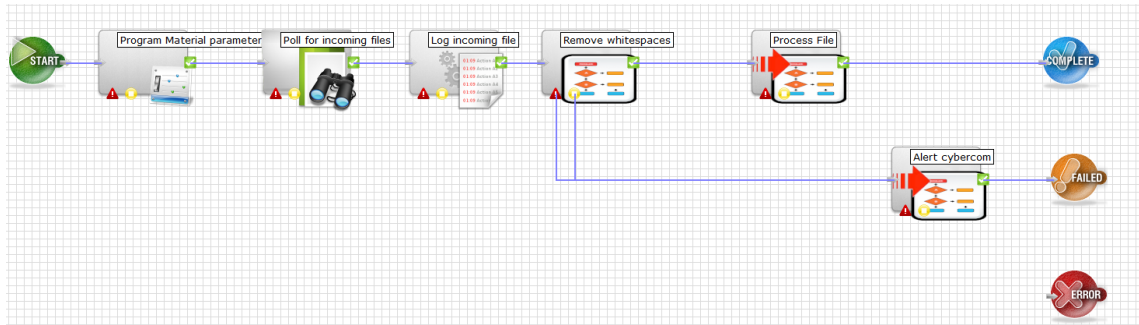
Käyttäjä täyttää tarvittavat tiedot lomakkeelle ja tämän jälkeen painaa "Submit", jonka jälkeen tiedosto siirtyy käsittelyssä eteenpäin.

Tähän astisen toteutuksen Orchestrator työnkulku on kuvien 19 ja 20 mukainen. Kuvassa 19 on yksittäisen tiedoston käsittelyn laukaiseva työnkulku.

Vasemmalta alkaen työnkulusta löytyvät seuraavat toiminnallisuudet

- parametrit
  - kohde- ja lähdehakemistot
  - sallitut tiedostotyypit
- uusien tiedostojen tarkkailija

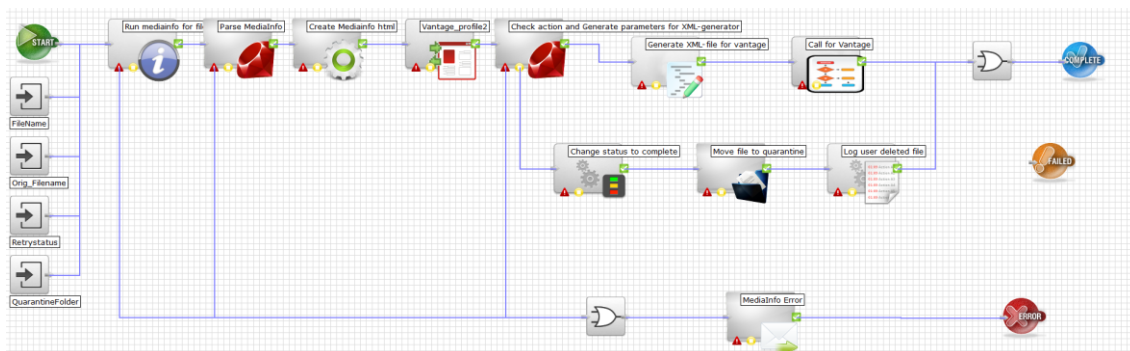
- logitus
- alityönkulku, joka poistaa saapuneista tiedostonimistä skandinaaviset merkit ja välilyönnit
  - o jos alityönkulku epäonnistuu, suoritetaan hälyttävä alityönkulku, joka lähettää virheilmoituksen käyttäjille
- alityönkulun kutsuminen joka suorittaa tiedoston varsinaisen käsittelyn



**Kuva 19: Käsittelyn laukaiseva työnkulku**

”Process File”- kohdasta kutstutaan kuvan 20 mukaista työnkulkua. Työnkulussa on vasemmalta lukien seuraavat toiminnot:

- aikaisemmasta työnkulusta välitetyt parametrit
  - o tiedoston nimi ja hakemisto
  - o kohdehakemisto
  - o ”retry status” parametri, joka kertoo onko tiedostoa yritetty käsitellä jo aikaisemmin
- mediainfo-sovelluksen kutsuminen
  - o jos tiedostosta ei pystytä lukemaan mediainfolla metatietoja, on tiedosto todennäköisesti rikki ja tästä ilmoitetaan käyttäjiä sähköposti-ilmoituksella.
- mediainfon tulosteen jäsentely
- luodaan mediainfon tiedoista HTML-sivu, josta käyttäjä näkee täydellisen tulosteen mediainfon palauttamista arvoista
- tehtävän luominen käyttäjälle
- käyttäjän syötteiden jäsentely ja suorituksen haarauttaminen
  - o joko jatketaan transkoodaukseen
    - Kutsutaan transkoodaavaa työnkulkua
  - o poistetaan tiedosto
    - poistetaan tiedosto ja kirjoitetaan logitiedostoon käyttäjän syötämä viesti.



Kuva 20: Tiedoston käsittely

### 6.3.2 Vantage-transkoodaus ja virheen käsittely

Saapuneesta tiedostosta muodostetaan symbolinen linkki (symlink) Vantagen näkemään levyjakoon. Mediainfon sekä käyttäjän tekemien valintojen perusteella Orchestrator käynnistää ääniraitojen analyysiin tarkoitetun Vantagen työnkulun Vantagen tarjoaman rajapinnan kautta. Ääniraitojen analysoinnin tarkoituksena on määrittää varsinaiselle transkoodaukselle välitettävät parametrit. Videotiedostojen ääniraitojen voimakkuudet voivat vaihdella suuresti riippuen materiaalityypistä. TV-lähetyskseen tarkoitetun materiaalin tulisi olla ääniraitojen voimakkuuksiltaan tasaista, jotta loppukäyttäjän ei tarvitse säätää vastaanotimesta voimakkuutta eri ohjelmien mukaan.

Vantagen analyysi palauttaa Orchestratorille XML-tiedostona ääniraitojen tiedot sekä arvot joiden mukaan suoritetaan ääniraidoille ekvalisointi varsinaisen transkoodauksen yhteydessä. Ekvalisoinnin tarkoituksena on tuottaa tasalaatuiset sekä äänenvoimakkuudeltaan yhtenevät ääniraidat.

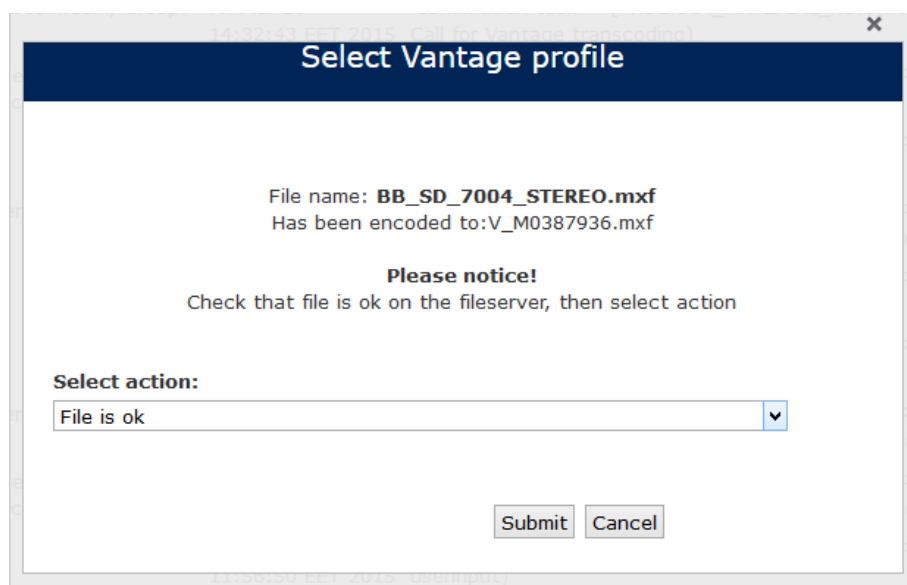
Analysoinnin jälkeen työnkulun suoritus jatkuu ja Orchestrator kutsuu varsinaisen transkoodauksen suorittavaa Vantagen työnkulkua. Vantagelle annetaan parametreiksi transkoodattava videotiedosto sekä audioanalyysistä saadun XML-tiedoston sisältö. Vantageen on määritelty useita työnkulkuja, joista jokainen on määritelty transkoodaamaan tietyn tyyppistä videomateriaalia. Vantageessa on omat työnkulut HD-tasoiselle sekä SD-tasoiselle videolle. Samoin lähdemateriaalin pakkaustapa vaikuttaa kutsuttavaan työnkulkuun. Valinta kutsuttavasta työnkulusta tehdään osana Orchestratorin työnkulkua mediainfo-sovelluksesta saatujen tietojen perusteella.

Vantage palauttaa Orchestratorille transkoodauksen tilatietoja, jotka esitetään käyttäjälle erillisellä työpöydällä. Tätä mahdollisuutta ei ollut aikaisemmassa toteutuksessa, käyttäjä sai ainoastaan ilmoituksen kun transkoodaus oli valmistunut tai virheen tapauksessa ilmoituksen virhetilanteesta. Nyt käyttäjällä on suora näkyvyys Vantagen käsittelyssä oleviin tiedostoihin ja tätä kautta materiaalinkäsittelijä pystyy arvioimaan kuinka kauan videotiedoston transkoodauksessa vielä kestää. Tämä mahdollistaa sen, että jos jonoon ilmestyy kiireellisellä prioriteetilla oleva tiedosto, niin käyttäjä pystyy priorisoimaan Vantagen jonotilan mukaan tiedoston pääsyn varsinaiseen transkoodaukseen.

Vantagen transkoodauksen valmistuessa käyttäjälle lähetään ilmoitus. Samoin, jos transkoodaus on päättynyt virheeseen, ilmoitetaan tästä käyttäjälle. Virhetilanteen tapauksessa tiedosto laitetaan uudelleen Orchestratorin jonoon ja siitä muodostuu vastaava tehtävä kuin uuden tiedoston tapauksessa. Tällöin tehtävän otsikkoon asetetaan tieto siitä, että tiedosto on jo kertaalleen ollut käsittelyssä.

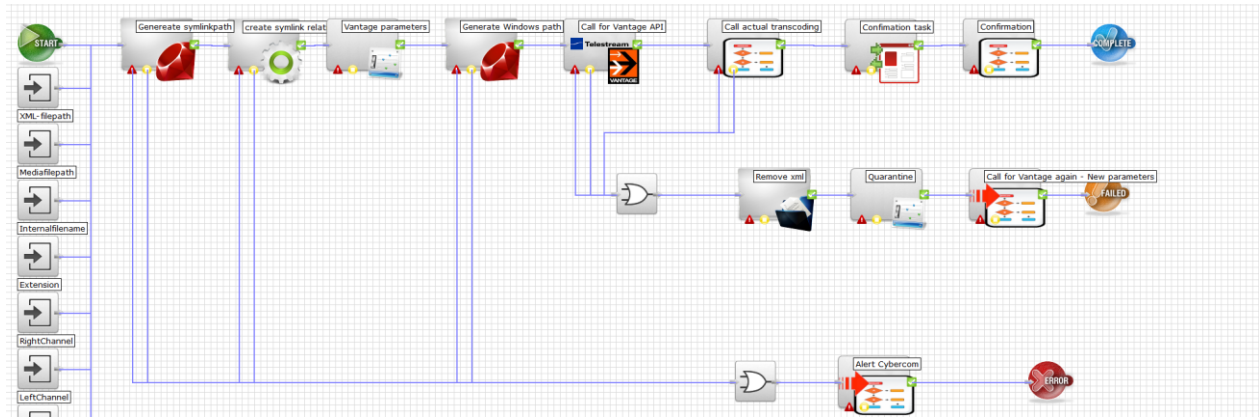
Jos Vantagen transkoodaus menee läpi, muodostuu tästä kuvan 21 mukainen varmistustehtävä käyttäjälle. Vantagen transkoodauksen jälkeen käyttäjän tulee käydä videopalvelimella varmistamassa, että lopputulos on haluttu. Käyttäjälle tarjotaan varmistustehtävässä mahdollisuus aloittaa työnkulku alusta. Yleisin virhetilanne milloin käyttäjä haluaa suorittaa transkoodauksen uudestaan on sellainen, jossa käyttäjä on määritellyt audioanalyysissä käytettävät ääniraidat väärin. Tällöin Vantage on käyttänyt audiokanavien ekvalisoinnissa vääriä kanavia ja lopputuloksessa puhutusta puheesta ei saada selvää.

Jos lopputulos on haluttu, käyttäjä vahvistaa tämän tehtäväikkunassa, jolloin alkuperäinen videotiedosto poistetaan ja yhden tiedoston suoritus on päättynyt.



***Kuva 21: Käyttäjälle näytettävä hyväksymistehtävä***

Kuvissa 22 ja 23 on esitettyä Orchestratorin työnkulun kaksi viimeistä vaihetta.

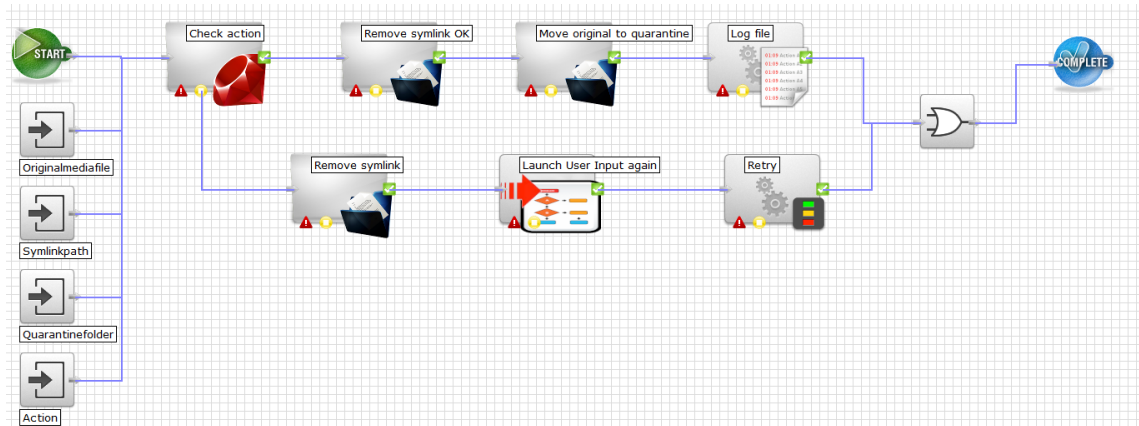


**Kuva 22: Tiedoston transkoodaava työnkulku**

Kuvassa 22 näkyy työnkulun vaiheet vasemmalta lukien:

- työnkulusta (kuva 20) välitettyjä parametreja. Listassa on muun muassa:
  - o käyttäjän syöttämä tiedoston sisäinen tunniste
  - o audiokanavien tiedot
  - o muita metatietoja tiedostosta, joita Vantage tarvitsee
- luodaan symbolista linkkiä varten polku
- luodaan symbolinen linkki Vantagen näkemään hakemistoon
- kiinteästi määritetyt Vantagen parametrit on talletettu erilliseen lohkoon
- vantage-transkooderit ovat Windows-palvelimia, jotka näkevät tiedoston Windows-muotoisessa hakemistossa: \\palvelin\hakemisto\tiedosto. Jotta tämä voidaan tuoda Vantagelle parametrinä, pitää se muodostaa erillisessä lohkossa. Eli muodostetaan Linux-tiedostopolun ja Windows-tiedostopolun välille yhteys
- kutsutaan Vantagen audioanalyysiä
  - o Palauttaa ulostulossa varsinaisessa transkoodauksessa tarvittavat ekvivalentitiedot
- kutsutaan varsinaista transkoodausta
  - o jos transkoodaus epäonnistuu, kutsutaan uudelleen kuvan 20 työnkulkua, jossa käyttäjä voi valita uudet ääniraidat transkoodaukseen.
- luodaan kuvan 21 mukainen hyväksymistehtävä
- kutsutaan käsittelyn viimeistä vaihetta





**Kuva 23: Käsittelyn viimeinen vaihe**

Viimeisessä vaiheessa suoritetaan seuraavat tehtävät:

- tarkastetaan käyttäjän valinta
  - jos tiedostossa ei ole ongelmia:
    - poistetaan symbolinen linkki Vantagen näkemästä levyjaosta
    - siirretään alkuperäinen tiedosto karanteeniin
    - luodaan logimerkintä, että tiedosto on transkoodattu onnistuneesti
  - jos tiedostossa on ongelmia (esim. audioraidat väärin)
    - poistetaan Symbolinen linkki
    - kutistutaan kuvan 20 työkulkua uudelleen ja kerrotaan käyttäjälle, että tiedosto on käsitelty jo kertaalleen ja että transkoodauksen lopputulos ei ollut haluttu.

Työkulun suorituksen päätyttyä onnistuneesti on materiaali valmiina käytettäväksi videopalvelimella. Lähetysautomaatiojärjestelmä käyttää videopalvelimella olevia tiedostoja kun ohjelma on tarkoitus esittää televisiossa.

## 7 YHTEENVETO

Uusi ympäristö on työn kirjoittamisen aikaan ollut käytössä noin vuoden verran. Ympäristöön on tänä aikana suoritettu jatkuvaa kehitystä, erityisesti uusien työnkulkujen osalta ja ympäristön laajentaminen uusiin käyttötarkoituksiin on ollut suoraviivaista ja helppoa.

Alkuperäinen tavoite siitä, että kaikki vanhat Bash-skriptit korvataan Orchestrator työnkuluilla, toteutui ilman suurempia ongelmia. Ongelmakohdiksi ovat muodostuneet lähinnä joidenkin Orchestratorin lohkojen hieman puutteelliset ominaisuudet. Nämä on kuitenkin kierretty kirjoittamalla itse omia lohkoja, joilla puutteet valmiiksi tarjotuissa toteutuksissa on saatu korvattua. Samoin yhteistyö Asperan kanssa on sujunut hyvin. Työnkulkujen toteutusten yhteydessä on törmätty muutamiin sovellusvirheisiin, joista olemme raportoineet Asperalle. Näihin on yleisesti saatu korjaukset viikon sisällä ilmoituksesta.

Suurimpana työnä toteutettu ohjelmamateriaalinkäsittely on työn kirjoitushetkellä ollut käytössä neljä kuukautta. Loppukäyttäjiltä saadun palautteen perusteella työnkulkuun ja erityisesti käyttäjille näytettäviin tehtäviin on tehty lähinnä kosmeettisia muutoksia, jotka tekevät käytöstä selkeämpää ja suoraviivaisempaa. Kustannussäästöjä on saatu aikaan, sillä QCToolsin ja muiden aikaisemmin käytössä olevien palvelimien käytöstä on voitu luopua uuden työnkulun käyttöönoton jälkeen.

Siirtojärjestelmässä ei ole ilmennyt vakavia ongelmia järjestelmän käytön aikana. Ongelmat ovat olleet perinteisiä palvelinylläpitoon liittyviä tehtäviä, jossa on jouduttu käynnistämään palvelin uudestaan esimerkiksi VMware-alustan tai verkko-ongelmien takia. Materiaalia on toimitettu FASP:n kautta aikavälillä 1.4.2014 – 1.4.2015 yhteensä noin 90 teratavua. Kokonaismäärä on tästä vielä suurempi, sillä FTP:n ja rsync-toimitusten määrästä ei ole saatavilla vastaavaa tilastoa.

Kokonaisuutena uusi ympäristö on vastannut sille asetettuja odotuksia. Tällä hetkellä tuotannossa on ajossa 30 eri työnkulkua. Vastaavasti muutamia työnkulkuja on sammuutettu, sillä niiden käyttötapaukset ovat olleet kertaluonteisia, esimerkiksi johonkin tapahtumaan liittyviä. Nämä työnkulut voidaan kuitenkin ottaa käyttöön kun niitä tarvitaan.

Järjestelmää jatkokehitetään jatkuvasti ja uusia työnkulkuja luodaan tarpeiden mukaan. Valmiiksi toteutettujen työnkulkujen pohjalta uusi, yksinkertaisen tiedostonsiirron toteuttava työnkulku on mahdollista toteuttaa kopiaimalla olemassa oleva työnkulku ja parametrisoimalla se uuteen tarpeeseen sopivaksi. Tämä on selvästi nopeampaa kuin Bash-skriptien kopiointi vanhassa ympäristössä.

Seuraavaksi tarkoituksena on korvata suuri määrä eräajoja Orchestratorin työkuluilla. Eräajoissa suoritetaan ajastetusti skriptejä, jotka muokkaavat esimerkiksi tietokannassa olevaa dataa erimuotoisiksi raporteiksi.

Sovelluksiin tulee aika-ajoin päivityksiä, jotka tuovat mukanaan uusia ominaisuuksia. Näitä ominaisuuksia otetaan käyttöön, jos niille nähdään tarvetta.

## LÄHTEET

- [1] James Kurose & Keith Ross, 2010, Computer Networking, Pearson, Boston, 888 s.
- [2] Fast and Secure protocol, Asperasoft, 2014, [WWW]. Viitattu 14.11.2014. Saatavilla: <http://asperasoft.com/technology/transport/fasp/>
- [3] J. Postel. User Datagram Protocol. RFC 768. IETF, 1980, 3s. Saatavilla: <https://www.ietf.org/rfc/rfc768.txt>
- [4] K. Zeilenga, Ed. Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map. RFC 4510. IETF, 6s. Saatavilla: <https://tools.ietf.org/html/rfc4510>
- [5] Active Directory, Microsoft Inc, 2014, [WWW], Viitattu 27.4.2015. Saatavilla: <https://msdn.microsoft.com/en-us/library/bb742424.aspx>
- [6] Cron jobs, Linux foundation, [WWW], Viitattu 20.4.2015. Saatavilla: [https://refspecs.linuxfoundation.org/LSB\\_2.0.1/LSB-Core/LSB-Core/sysinit.html](https://refspecs.linuxfoundation.org/LSB_2.0.1/LSB-Core/LSB-Core/sysinit.html)
- [7] RedHat Enterprise Linux, Redhat, [WWW], Viitattu 28.12.2015. Saatavilla: <http://www.redhat.com/en/resources/red-hat-enterprise-linux-server>
- [8] VMware homepage, VMware Inc., [WWW], Viitattu 18.12.2015. Saatavilla: <http://www.vmware.com/>
- [9] T.Ylönen, C. Lonvick, Ed. The Secure Shell (SSH) Transport Layer Protocol. RFC 4235. IETF, 2006, 31 s. Saatavilla: <https://tools.ietf.org/html/rfc4253>
- [10] IPTables, Centos Community, [WWW], Viitattu: 20.4.2015. Saatavilla: <http://wiki.centos.org/HowTos/Network/IPTables>
- [11] What is vantage – Telestream, [WWW], Viitattu 10.5.2015. Saatavilla: <http://www.telestream.net/vantage/overview.htm>
- [12] RPM – RPM.org, [WWW], Viitattu 28.4.2015: Saatavilla: <http://www.rpm.org/>
- [13] Connect Server installation, Asperasoft, 2014, [WWW], Viitattu: 10.12.2015. Saatavilla: [http://download.asperasoft.com/download/docs/entsrv/3.5.2/cs\\_admin\\_linux/webhelp/index.html](http://download.asperasoft.com/download/docs/entsrv/3.5.2/cs_admin_linux/webhelp/index.html)
- [12] MySQL Community Edition – MySQL.com, [WWW], Viitattu 10.5.2015. Saatavilla: <https://www.mysql.com/products/community/>